OPTIMAL CONTROL OF CONSTRAINED HYBRID DYNAMICAL SYSTEMS: THEORY, COMPUTATION AND APPLICATIONS

A Dissertation
Presented to
The Academic Faculty

by

Usman Ali

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2016

OPTIMAL CONTROL OF CONSTRAINED HYBRID DYNAMICAL SYSTEMS: THEORY, COMPUTATION AND APPLICATIONS

Approved by:

Professor Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yorai Wardi, Co-Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor David Taylor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Erik Verriest
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Evangelos Theodorou
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: July 22, 2016

To my father, Firdaus Rahim and my mother, Tauheed Begum.

ACKNOWLEDGEMENTS

Being at Georgia Tech for the past six years as a Master's leading to a PhD student has been a truly wonderful experience. I got a chance to come in contact with the best of the best, who contributed to my personal and academic growth in a number of ways. The successful completion of the work presented in this thesis is indebted to a number of people and while it is impossible to recount and acknowledge every one individually, the important ones are pointed out in the lines below.

I would like to begin with thanking my advisor Professor Magnus Egerstedt, the chief architect behind the work presented in this thesis and a source of great inspiration. I am ever grateful to him for his patience with me over the years to grow as an independent researcher. His clear thinking and effective articulation provided a great intuition in understanding concepts that are often hidden behind the mathematical rigor. Whenever I needed him for help, he was there. Apart from his effective advising over the years, his keen insight, his highly productive time management, his discipline and his drive for impact have been and will continue to be a source of great inspiration for me. I am also highly grateful to my co-advisor, Professor Yorai Wardi who spent a lot of time and energy on me and made this thesis possible. Besides being a great advisor, my discussions with him on a spectrum of topics ranging from soccer and food to politics and history and his great company would always be remembered.

I am so much thankful to my mother, who had to bear without me in the home country for a period of six years, to see the fulfillment of my aspirations and to see the dream of my late father to achieve academic excellence come true. The emotional support that I received from her during the lows and highs of my PhD program has been truly amazing. When things would come hard to pass by, her prayers would

always do the miracle for me. The same goes for my siblings who have been long waiting for this moment and with whom I celebrated every small success during this journey.

The members of my Lab, Georgia Robotics and Intelligent Systems Laboratory (GRITS), have been truly wonderful people and working with them has been a truly joyful experience. Thank you all for your valuable input on my work from time to time and being a great company. I would also like to take this opportunity to thank some of our collaborators, in particular Professor Yasamin Mostofi at University of California, Santa Barbara with whom some of the work on co-optimization in this thesis was done. I am grateful to Professor Todd Murphey from Northwestern University, Professor Erik Verriest from Georgia Tech for valuable discussions on the dwell time problem presented in this thesis.

Life would have been quite boring in the absence of my family, had it not been for the truly wonderful friends and roommates I had over the years from the Pakistan Fulbright community and I am grateful to all of them. In particular, I would like to thank Faisal Kakar, Abdul Basit, Ali Ahmed, Haider Ali and Usman Gul for their camaraderie. The memory of good times spent with them will continue to haunt me over the years to come.

Towards the end, I would like to acknowledge the U.S. Department of State and Higher Education Commission of Pakistan for providing me with Master's leading to PhD scholarship under *Fulbright* Program. The people at United States Education Foundation in Pakistan (USEFP) and Institute of International Education (IIE), who supervise this program, did a fabulous job in making various bureaucratic tasks seamless and providing a number of enrichment opportunities, apart from those provided by the institute. During my stay here, I came in contact with people from a number of nationalities and cultures which provided a great opportunity for mutual understanding and thus furthering the objective of Fulbright. Thanks!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Hybrid dynamical systems arise in a number of application areas such as power converters, autopilots, manufacturing, process control, hybrid cars, mobile and humanoid robotics etc., to name a few and as such the optimal control of these systems has been an area of active research. These systems are characterized by two components: subsystems (modes) with continuous or discrete dynamics and a switching law which determines which of these subsystems is active at a given time. While in theory, we can switch infinitely many times between different modes in a finite amount of time, physical systems need to spend some minimum time in a mode before they can switch to another mode due to mechanical reasons, power constraints, information delays, stability considerations etc and must spend some minimum amount of time in a mode before they can switch to another mode. This minimum time is known as the *dwell time*, a term first used in the context of stability of hybrid systems, and the optimal control of hybrid systems under these constraints is the main focus of this thesis.

The presence of the dwell time constraints raises interesting theoretical and computational questions which are addressed in thesis. We consider the general hybrid optimal control problem subject to dwell time constraints thereby establishing necessary conditions for optimality and develop numerical schemes to compute solutions to these problems and prove their convergence. Any physical system that switches is subject to dwell time constraints, small or large, and thus amenable to our framework. To demonstrate, however, the generality and thus wide applicability of our results, we consider the application to an interesting problem in Precision Agriculture, namely the problem of optimal pesticide scheduling and present a case study to demonstrate

the application of our methodology.

In this thesis, we also consider a class of constrained hybrid optimal control problems inspired by problems in power aware mobile robotic networks that are subject to various constraints on inputs and states. In particular, we consider the problem of jointly minimizing motion and communication energy in power aware mobile robotic networks required to perform various co-ordinated tasks such as the transmission of given amount of data to a remote base station under time and resource constraints and where the robot decision variables are acceleration (continuous), for controlling the motion of the robot and spectral efficiency (discrete), catering to data transmission requirements. Framing this co-optimization problem as a constrained hybrid optimal control problem in the general setting and subsequently solving it using efficient algorithms is another main topic of this thesis. This problem, like any other hybrid optimal control problem, is also subject to dwell time constraints, signifying the importance of the dwell time problem addressed in this thesis. We present numerous application scenarios to demonstrate the utility of our framework.

Finally, we propose a multiple shooting based gradient descent techniques to solve a class of complex optimal and hybrid control problems with large time horizons, which otherwise are hard to solve due to numerical problems arising from instability issues associated with the state or co-state equation. The two point boundary problem resulting from solving the optimal or hybrid optimal control problem is transformed into an equivalent optimal control problem over extended states comprising of the original state equation and the costate equation and then solved. Again, the results here are general and we demonstrate the effectiveness of our method by considering its application to solving large multi-agent co-optimization problem in power aware mobile robotic networks.

CHAPTER I

INTRODUCTION

Hybrid dynamical systems arise in a number of application areas such as mobile and humanoid robotics, automotive engine control, manufacturing, process control, power converters, hybrid cars etc; see the survey [8] and references therein. As such, the optimal control of these systems has been an active area of research. These systems are characterized by two components; subsystems with continuous or discrete dynamics and controlled or autonomous switching between these subsystems and this allows for modeling of a large number of natural phenomena and complex control systems. Due to its scope, the field has attracted researchers from mathematics, control engineering and computer science who have made numerous theoretical and computational contributions over the entire spectrum of control theory, ranging from stability analysis and optimal control to safety and verification. The objective of the work presented in thesis is to solve a class of constrained hybrid optimal control problems by deriving necessary conditions of optimality for them and devise numerical schemes for their efficient computation. We demonstrate the utility of our results for the general hybrid optimal control problem by considering their application to problems in power aware mobile robotic networks and precision agriculture.

The control parameters for hybrid systems consists of a switching law and a continuous external input. The switching law, which governs which subsystem is active at a given instant of time is defined using control variables that describe the sequence of modes and the duration of each mode and commonly referred to as a mode schedule when considered together. The subsystems are also commonly referred to as the modes of the system in literature [76]. A common optimal control problem

is then to minimize a cost functional defined on the state trajectory and the control inputs as a function of the switching law and continuous input and variations thereof [10, 12, 16, 22, 25, 36, 56, 65, 66, 69, 77, 80, 82].

While in theory, we can switch infinitely fast between different modes in a finite amount of time, most physical systems have to spend some minimum time in a mode before they can switch to another mode due to mechanical reasons, power constraints, information delays, stability considerations e.t.c. This minimum time is known as the dwell time of a mode, a term previously used in the context of stability of switched linear systems [41, 51]. This important and practical constraint has not received attention in the research on optimal control of hybrid dynamical systems and deriving optimality conditions and numerical schemes for computing optimal solutions to the general hybrid optimal control problem under dwell time constraints is one of the main focus of this research. The presence of the dwell time constraints raises interesting theoretical and computational questions which are addressed in thesis. Any physical system that switches is subject to dwell time constraints, small or large, and thus amenable to our framework. To demonstrate, however, the generality and thus wide applicability of our results, we consider the application to an interesting problem in Precision Agriculture, namely the problem of optimal pesticide scheduling and present a case study to demonstrate the application of our methodology.

After introducing the requisite background in chapter 2, we present the problem of optimal control of hybrid dynamical systems under dwell time constraints in chapter 3 where the problem is formally introduced in a general setting and necessary conditions for optimality are presented. The presence of the dwell time constraint in the hybrid optimal control problem raises a number of interesting theoretical questions and presents significant computational challenges. To compute solutions to hybrid optimal control problems that are numerically tractable, most optimal control techniques tend to compute solutions that are locally optimal. While we have an understanding

of local optimality for hybrid optimal control problems, it breaks down when dwell time constraints are present. Hence, developing a notion of local optimality for such constrained hybrid optimal control problem is the first main contribution of this thesis. To develop this notion, we investigate the topology of optimization space with and without the dwell time constraints and define a notion of local minima for such problems. Having understood the notion of locally optimal solution, an optimality function is then proposed to characterize the necessary optimality conditions for our dwell time problem. Our this contribution appeared in [5] where we also presented a conceptual algorithm for computing solutions to the dwell time problem in terms of the above defined optimality conditions.

Computing analytical solutions to optimal control problems is hard and hence once needs to resort to some sort of algorithms to compute the optimal controls and hence chapter 4 deals with developing effective computational techniques for computing the optimal controls to the dwell time problem. The presence of dwell time constraints further breaks down the continuity properties present otherwise which makes the direct application of methods based on variational calculus ineffective. In chapter 3, we present a two step strategy to compute the optimal controls by dividing the problem into two stages. In the first stage, the mode sequence is kept fixed and optimization is performed over switching times only. In the second stage, the mode sequence is updated by the introduction of a new mode to the sequence for duration of dwell time at a point that gives maximum decrease in cost. While the first stage is solved effectively, the second stage relies on the explicit computation of cost differential resulting from the introduction of a new mode at every time instant for the duration of dwell time which is highly inefficient, and is the reason we refer to it as a conceptual algorithm. In chapter 4, we present an effective method for updating the mode sequence by utilizing information on mode insertion gradient [25] which has its roots in the Maximum Principle and is readily computable. This entails a

deep investigation into the properties of mode insertion gradient and it is undertaken in this chapter. We study the geometric properties of the mode insertion gradient and present its connection with the results from Maximum Principle. We present a number of results on the properties of mode insertion gradient and formally present our dwell time algorithm and prove the convergence of the algorithm to the stationary points of the optimality function introduced in chapter 3. A number of illustrative examples, including linear and nonlinear systems are provide to demonstrate the application of our approach for solving hybrid optimal control problems subject to dwell time constraints. Details of the work presented in chapter 3 and chapter 4 can also be found in [4].

In the aforementioned work, we have assumed that each mode has a fixed dwell time. For a mode to exist in the sequence, it must respect this dwell time constraint and this results on constraints on the adjacent switching times only. In chapter 5, we extend the switch time optimization part of our dwell time problem to the more general case where the modes of the system need to satisfy the dwell time constraints not only with respect to adjacent modes but also other modes in the sequence and the dwell times for each mode need not be the same. These type of problems arise for example in precision agriculture, where there are constraints on the time and schedule of use of different pesticides for pest risk control. We will consider the specific case of blueberry farming to demonstrate the impact of the generality of our results derived for hybrid dynamical systems for solving problems in diverse areas as agriculture. The work presented in this chapter also appeared in [45] and a journal version has been submitted to Transactions on Cyber Physical Systems (TCPS) [3]

We next consider a class of constrained hybrid optimal control problems inspired by problems in the area of power aware mobile robotic networks where the objective is to balance the motion and communication energy of robots in a network subject to various constraints on the input and state. This problem of co-optimizing motion

and communication energy in multi-agent robotics in fading environments has the structure of a hybrid optimal control problem and formulating this problem as an optimal control problem and subsequently solving them using efficient algorithms to achieve co-ordinated tasks is another focus of this work. The control parameters for this problem are acceleration, which is continuous, and spectral efficiency which is a discrete and the problem is hybrid both in the objective function and dynamics with a number of constraints on the trajectory and controls. We solve the co-optimization problem first with the assumption that the spectral efficiency is a continuous variable and later on we present a method for tackling the problem with switching dynamics in chapters 6 - 8. The framework is developed in the general setting and application examples are provided to validate the results.

In chapter 6, we present an optimal control approach for solving a motion and co-optimization problem in realistic fading environments. We consider the problem of a robot that is tasked to transmit a fixed number of bits while moving a long a fixed path from some starting point to a given final point in a fixed amount of time. The quality of channel is estimated based on few channel measurements a priori using the channel prediction framework of [32]. The co-optimization problem is framed as an optimal control problem amenable to the use of maximum principle and hence the use of a Hamiltonian based algorithm for effective solution. We use realistic models for motion and communication costs and system dynamics and the application of our method is demonstrated by presenting simulation results. The work presented in this chapter was also published in [6].

The co-optimization problem presented in chapter 6 assumes that we have channel measurements available at few unvisited locations a priori from which the channel quality at unvisited locations can be predicted using the channel prediction framework of [32]. Moreover, the path the robot has to traverse between the source and destination is fixed both of which are quite restricting assumptions. In chapter 7,

we extend the co-optimization problem to the case where the robots has to predict the channel on the go and solve the optimization problem in real time like fashion as it gets better understanding of the channel. This requires the usage of efficient algorithms and this will be addressed in chapter 7. Details of the work on the online co-optimization problem in the plane can also be found in [1, 2].

The power aware problem considered in chapters 6 and 7 considers the spectral efficiency as continuous, while in reality, the spectral efficiency takes discrete values and the problem is thus hybrid in nature. In chapter 8, we pose the co-optimization problem presented in chapter 6 in the framework of a constrained hybrid optimal control and solve the problem subsequently using relaxation based approach and explain the application of our method by considering its application to an example problem.

The spectral efficiency cannot switch instantly and hence the hybrid co-optimization problem like any other hybrid optimal control problem is also subject to dwell time constraints. However in this case, the dwell times are in milliseconds, compared to the problem of optimal pesticide scheduling where the dwell time constraints are in days. Since any physical system that switches exhibits dwell time constraints, small or large, a number of application problems were candidates for the application of our framework. We chose the problem of optimal pesticide scheduling in precision agriculture to demonstrate the generality and hence wide applicability of our results for the general constrained hybrid optimal control problem.

When solving optimal or hybrid optimal control problems over a large horizon, difficulties arise due to instability issues associated with the state equations or co-state equations which adversely affects the convergence properties of the Hamiltonian algorithm. Chapter 9 presents a new multiple shooting based technique for solving complex optimal and hybrid optimal control problems. While the method presented is general, we demonstrate its strength by considering the co-optimization problem

for large team of robots in power aware networks, that are otherwise very hard, if not impossible to solve. In our method, the two point boundary value problem resulting during the process of solving optimal control problem is solved by setting it up as a new optimal control problem. An alternate cost is defined on the terminal conditions and the cost is minimized subject to the dynamics described by differential equations of boundary value problem and the cost is minimized over the initial condition. This work was also published in [7]

Chapter 10 concludes the thesis and present directions for future research. We next begin with a brief overview of hybrid optimal control problem and some of the existing results in the literature pertinent to this thesis.

CHAPTER II

HYBRID OPTIMAL CONTROL - AN OVERVIEW

Hybrid dynamical systems (HDS) have been extensively investigated in the recent years since such systems arise in a number of application areas such as mobile robotics, power converters, manufacturing, autopilots, process control, automotive industry, chemical processes, traffic control, communication networks, biomedical applications, humanoid robotics, heating and cooling systems, modeling natural phenomena and the list goes on. The special issue [8] gives a good overview of the scope of Hybrid Dynamical Systems. Before proceeding towards the main topic of this thesis, we give a brief description of the different representations for Hybrid Dynamical Systems useful in the context of hybrid optimal control and therefore this thesis, and we refer the reader to [75] for detailed description hybrid dynamical systems. We then describe the common hybrid optimal control problems and conclude with a brief overview of the existing results on the optimal control of hybrid dynamical systems relevant to this thesis.

2.1 Representation of Hybrid Dynamical Systems

In hybrid dynamical systems, the switching between different subsystems can be autonomous or controlled depending on whether the switching is caused by system passing through different boundaries or forced by control input. An example of the first type is that of a bouncing ball while gear shifting in automobiles is an example of the latter. Since we are interested in the optimal control of these systems, we will be considering only controlled hybrid dynamical systems in this thesis and we refer the reader to [15] which provides a rigorous description of the hybrid dynamical systems and explains the different types of hybrid dynamical systems with examples.

As mentioned briefly in the introduction, the control parameters for hybrid dynamical systems are characterized by a switching law which determines which subsystems is active at a given time and a continuous external input that drives each subsystem. Mathematically, more than one representations are possible to describe these systems, each with its own merits. A bijection however exists between all these representations and are used interchangeably in the literature and also in this thesis. We briefly describe each of these representations below.

2.1.1 Switch Signal Representation

The dynamics of continuous time HDS can be represented as

$$\dot{x}(t) = f_{v(t)}(x(t), u(t)), \quad x(0) = x_0, \tag{1}$$

where $x \in R^n$ is the state of the system, x_0 is the initial condition, $u \in R^m$ denotes the continuous control input, $v : [0,\, t_f] \to \mathcal{Q} = \{1, \ldots, N\}$ is a discrete control input that specifies the active subsystem (also referred to as switching law) at any given time and $f_q : R^n \times R^m \to R^n$ is the control vector field denoting the dynamics of subsystem $q \in \mathcal{Q}$ that is continuously differentiable in x and locally Lipschitz continuous in u. An example of this representation is shown in Fig. 1

An alternate representation using switching signals is also sometimes used as follows. Let $f_i(x(t), u(t))$ denote the dynamics of each individual subsystem i and let

$$v_i : [0,\, t_f] \to \{0, 1\}$$

be the control associated with this mode such that

$$\sum_{i=1}^{N} v_i(t) = 1 \quad \forall \quad t \in [0,\, t_f],$$

then the dynamics in equation (1) can be equivalently written as

$$\dot{x}(t) = \sum_{i=1}^{N} f_i\big(x(t), u(t), t\big)\, v_i(t), \quad x(0) = x_0, \tag{2}$$

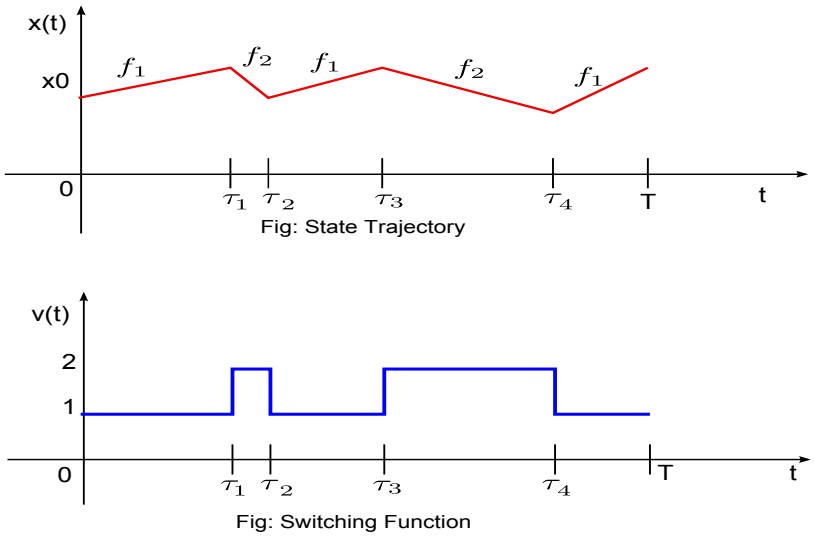


Figure 1: An example of hybrid dynamical systems representation using switching signal form. The active mode at any time instant is given by $v(t)$. The transition between active subsystems happen at the instant when $v(t)$ switches its value. The evolution of state trajectory with time as a function of the switching signal is shown in red.

The switching control in this representation is then the vector $v = (v_1, v_2, \ldots, v_N)^\top$ that maps from

$$v : [0,\, t_f] \to \{0, 1\}^N$$

where N is the total number of subsystems that constitute the hybrid system and u is the continuous external input to each subsystem. The equivalence between the two representations is quite straightforward.

2.1.2 Mode Schedule Representation

The switching law $v(t)$, which governs which mode is active at a given instant of time is often defined using control variables (σ, τ) that describe the sequence of modes (discrete variable) and the duration of each mode (continuous variable) respectively and commonly referred to as mode schedule $\xi = (\sigma, \tau)$ when considered together [10]. The switching signal $v(t)$ corresponds to the mode schedule $\xi = (\sigma, \tau)$ through a bijection as follows. Let $\tau = (\tau_1, \tau_2, \ldots, \tau_n)'$ be the vector corresponding to monotone

increasing transition times in $v(t)$ in interval $[0, t_f]$. Let $\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(n+1)\}$ denote the mode sequence, where $\sigma(i) = v(\tau_i)$ is an element of the discrete set $\mathcal{Q}$, so $\sigma \in \mathcal{Q}^{n+1}$. If we define $\tau_0 = 0$ and $\tau_{n+1} = t_f$, then $v(t) = \sigma(i)$ for all $t \in (\tau_{i-1}, \tau_i]$ and $i = 1, 2, \ldots, n+1$ and $\xi = (\sigma, \tau)$ is the corresponding mode schedule. We define the length of mode schedule ξ to be the same as the length of mode sequence σ. For schedule of length $n+1$, (3) can be written in the form

$$\begin{aligned} \dot{x}(t) &= f_{\sigma(i)}(x(t), u(t)) \quad \forall \quad t \in (\tau_{i-1}, \tau_i], \\ x(0) &= x_0, \end{aligned} \tag{3}$$

and $i = 1, \ldots, n+1$. The bijection is explained in Fig. 2 for the case of hybrid system with two subsystems.

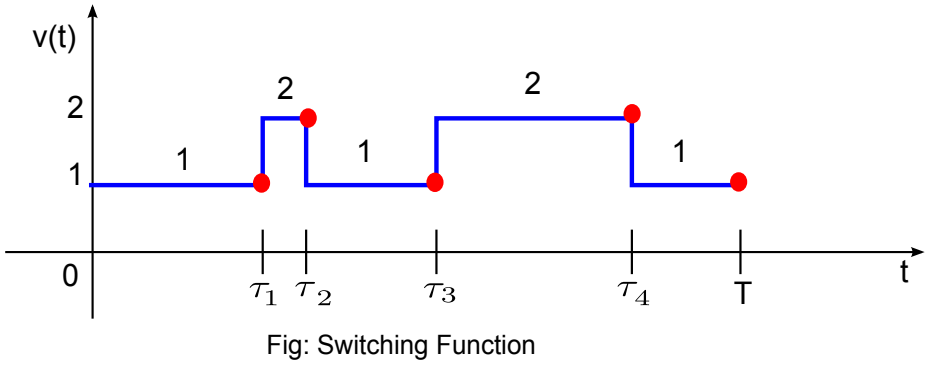


Fig: Switching Function

Figure 2: Illustration of bijection between the switching signal representation and mode schedule representation. The small red circles indicate the switching function is assumed to be left continuous.

2.2 Hybrid Optimal Control Problem

In optimal control problems, one seeks to find a control signal over the set of admissible controls that minimizes or maximizes a certain desired performance criterion while respecting the physical constraints of the process or system. Most performance criterion can however be abstracted away into a common functional form which we describe next.

Let $L_q : R^n \times R^m \to R$ be a continuously differentiable function in x and locally Lipschitz continuous in u for each $q \in \mathcal{Q}$ and let $\phi : R^n \to R$ be the cost on the terminal state that is continuously differentiable in x. The common optimal control problem (and variations thereof) considered in hybrid dynamical systems is then to minimize the cost functional

$$J = \int_0^{t_f} L_{v(t)}(x(t), u(t))\, dt + \phi(x(t_f)) \tag{4}$$

as a function of the control pair (u, v) subject to dynamical constraints (1) and constraints on the input and state. It is possible to write the cost functional (4) using the other two representations as well and we will them interchangeably in this thesis.

The general hybrid optimal control problem is hard to solve and as a result, initial investigation considered optimization over a subset of the control variables. The four different type of hybrid optimal control problems investigated in the literature are

1. *Switch Time Optimization*: Here optimization is performed over the switching times (τ) between modes only while keeping the sequence of modes fixed for autonomous systems.

2. *Mode Schedule Optimization*: Optimization is performed over the the switching times τ as well as the sequence of modes σ i.e. mode schedule $\xi = (\sigma, \tau)$ for autonomous systems.

3. *Continuous-Control Optimization*: The switching control $v(t)$ or alternately ξ is absent in these problems and the optimization is performed over the the continuous external $u(t)$. The switching between different modes happen as the system passes through switching surfaces defined by constraints on the state.

4. *Full Optimization*: The cost functional (4) is minimized as a function of mode schedule $\xi = (\sigma, \tau)$ as well as the continuous external input $u(t)$ for controlled systems.

The above hybrid optimal control problems have been approached using both of the major optimal control tools i.e Dynamic Programming [11] and Maximum Principle [58] and a number of theoretical and computational contributions have been made. We next give a brief overview of some of the main results relevant to this thesis while we refer the reader to [46] for a good overview of the two principles.

2.3 Overview of Existing Results

The optimal control of these systems has been an active area of research for the past two decades and results of theoretical and numerical significance have been derived. While the length of this document precludes an exhaustive survey of these results, the survey below provides an overview of the major contributions in the area that are relevant to our proposed research. This work aims to consider the application of our theoretical results to solve problems in power aware mobile networks, and to this end, we will also provide a brief literature overview of the problems in this area.

On the theoretical side, a number of results on necessary conditions for optimality have been derived for both autonomous and controlled systems. In [16], a general framework for the hybrid optimal control problem was formulated and existence of optimal controls discussed in light of dynamic programming. Variations of maximum principle were derived for necessary conditions on optimality in [55,56,61,69,70]. The authors in [39, 40, 59] approached the problem using convex dynamic programming to approximate the optimal feedback control laws by discretization of state space for solving Hamilton Jacobi Bellman equations. In [66], necessary optimality conditions were derived using maximum principle for a fixed mode sequence while the results were extended to more general mode scheduling problem in [18, 65, 67].

On the computational side, a number of algorithms have been developed over time to compute the optimal controls utilizing the necessary optimality conditions derived on the theoretical front. Early algorithms focused on hybrid systems with fixed mode

sequence and the optimization was performed over switching times only since the control variable τ is continuous which makes it amenable to nonlinear programming techniques and consequently many algorithms were first developed to solve this switch timing optimization (STO) problem. In [26, 27, 79, 80], expressions were derived for the cost gradient to be subsequently used in steepest descent algorithms. In [20, 44], the second derivative of cost was used in addition to the first derivate and it was shown that these second order methods yield faster convergence convergence to the optimum when compared to the first order methods in [26, 27, 79, 80]. The authors in [33] considered autonomous STO problem for the case of affine systems with state jumps and considered cost on switchings in addition to quadratic cost on the state, and showed that the optimal control for such problems takes the form of feedback law. In [21], the autonomous switch time optimization problem was solved for linear time-varying systems and a quadratic cost. The authors in [27] considered the problem with state estimation based on partial output information. The case of online optimization of switching times was dealt with in [24] which is helpful in real time applications as the system gathers more information on the go. In [81, 82], the results of [80] were extended to the case of *controlled* STO problems.

In case of the mode scheduling optimization problem, the control variable has a continuous component τ as well as discrete component σ which renders the problem significantly complex. A number of computational strategies have been developed to solve the scheduling problem autonomous and non-autonomous scheduling problem that can be broadly classified into three categories: algorithms based on needle variations [10, 36, 76], zoning and location algorithms [18, 19, 65], and relaxation based methods [12, 22, 38]. The first set of methods solve the problems in two stages iteratively. At the highest level, the mode sequence is modified by inserting a new mode while at the lower level the switch time optimization problem is solved. In [10], at

every iteration, a new mode is inserted to the existing sequence based on insertion gradient at a point. The authors in [36] extends this by considering external continuous input and adding constraints on the states. The inefficiencies associated with inserting a single mode to the sequence at a time in [10,36] are addressed in [76–78] which adds multiple modes to the mode sequence at every iteration. The authors in [18, 19, 65] exploited the geometric nature of the scheduling problem and introduced the notion of optimality zones and subsequently incorporated their computation in their hybrid maximum principle algorithm [67]. In [12], the discrete dynamics were embedded into a continuous time framework and techniques of optimal control were employed to solve the problem. It was shown that set of trajectories of switched systems are dense in the trajectories of the continuous time system. A projection operator approach was proposed in [22] for solving the mode scheduling problem and necessary conditions for optimality derived in this setting. The relaxed controls are projected onto the constrained space during each step of algorithm. Recently, a Hamiltonian based relaxed optimal control approach has been employed by us in [38] for class of hybrid systems that are affine in control and convex in control in the cost. The solution to the switched system is then obtained by using pulse width modulation and the algorithm is shown to have nice convergence properties. All these algorithms are shown to be provably convergent to optimal solution.

The solutions to optimal mode scheduling problem can be chattering i.e. switch infinitely many times between different modes in a finite amount of time while most physical systems have to spend some minimum time in a mode before they can switch to another mode due to mechanical reasons, power constraints, information delays, stability considerations etc. This minimum time is known as the dwell time of the mode, a term previously used in the context of stability of switched linear systems [41,51]. This important and practical constraint has not received attention in research on optimal control of switched systems. Motivated by this, some of our preliminary

work in [4, 5] considered the mode scheduling problem under dwell time constraints and derived necessary optimality conditions and proposed an algorithm to solve the problem. In [3, 45], we considered the problem of more generalized constraints on the mode sequence and considered the application to solve an interesting problem of optimal pesticide scheduling in precision agriculture.

In this thesis, we also consider the problem of constrained hybrid optimization inspired by problems in the area of power aware mobile robotic networks and to this end, we will next provide a brief overview of the work done in this area pertinent to this thesis.

2.4 Power Aware Mobile Networks

The field of communication-aware robotics concerns power management by mobile robotic systems which have to perform coordinated tasks over communication networks with limited energy resources. The robots are most often DC powered and do not have access to power outlet before completion of task. This makes the energy available to perform various tasks such as sensing, motion, communication, computation a limited resource. Experiments have shown that the main sources of energy consumption are motion and communication. While individual optimization of motion and communication energies have received great attention in literature, joint optimization of motion and communication energy have started to attract attention only recently [2, 6, 7, 28, 34, 43, 54, 71, 83–86] and the term co-optimization is used to refer to these type of problems. In this thesis, we consider the problem of motion and communication co-optimization in fading environments using realistic models for motion and communication energy in an optimal control framework and then present how the problem is hybrid in dynamics in addition to being hybrid in objective and address the problem from the perspective of hybrid optimal control. We next give a brief overview of some of the main results in the above references pertinent to this

thesis.

Ref. [34] considers motion planning for mobile relays so as to minimize their transmission energy. While it does not consider explicit motion costs, the provided simulation results show that the computed paths come close to optimizing the combined total communication/motion energy. Ref. [71] extends the algorithm in [34] to maximize, on line, the lifetime of wireless sensor networks by considering transmission costs as well as motion-energy costs. Ref. [54] presents an approximate path planning algorithm based on Dijkstra's algorithm for minimizing the combined motion and communication energy. Ref. [28] proposes an approach for minimizing the motion-communication energy costs in a relay network. First the robots' trajectories are computed for minimum motion energy, then the transmission schedule is computed to minimize the communications cost. Ref. [86] considers the problem of combined energy minimization in the framework of LQR model-predictive control and proposes a distributed algorithm for it, and [85] follows a similar approach for maximizing the lifetime of wireless sensor networks.

Dynamic, optimal-control formulations of the co-optimization problem were proposed in [6, 43, 83, 84]. Refs. [83, 84] consider a discrete-time formulation and apply nonlinear-programming techniques, and Ref. [6] considers a continuous-time setting and uses a specialized algorithm. We point out that the models for transmission power and motion power in [83, 84] are precise and realistic, whereas the algorithm used in [43] is applied to a simplified model but displays fast convergence. The two approaches are combined in our work in [1,2,6] to result in a fast-convergent algorithm for a continuous-optimal control problem formulation under realistic assumptions and solving the co-optimization problem in realistic environments while in [7] we solve a complex multi-agent problem over large horizon using a new multiple shooting algorithm and these works will be presented in the later part of this thesis.

The problems considered in our work [1, 2, 6], while hybrid in objective are also

hybrid in dynamics but was solved with the assumption of continuous dynamics and as a part of this thesis, we will frame the problem as a hybrid optimal control with switching dynamics and subsequently solve the problem later in this work. Moreover, like any other switched system, the hybrid co-optimization problem is also subject to dwell time constraints and thus amenable to the dwell time framework presented in this and this is the common thread between the otherwise two disparate constrained hybrid optimal control problems considered in this thesis.

While we consider the application of our results to solve the problem of optimal pesticide scheduling in precision agriculture, our framework can be extended to solve the co-optimization problem subject to dwell time constraints as well. In fact, a number of other interesting problems can be addressed in our dwell time framework and thus were candidates for the application of our results since we consider the problem in the general setting. However we choose the problem of optimal pesticide scheduling to demonstrate the application of our results in a diverse area as agriculture, thereby highlighting the impact of the generality and hence wide applicability of our results. We begin with a formal introduction to the dwell time problem in the next chapter.

CHAPTER III

OPTIMAL CONTROL OF HYBRID DYNAMICAL SYSTEMS UNDER DWELL TIME CONSTRAINTS

The problem of determining optimal switching law for systems that switch between different modes has been extensively investigated in recent years since such systems arise in a number of application areas and results of theoretical and computational significance have been derived. While in theory, we can switch infinitely many times between different modes in a finite amount of time, most physical systems have to spend some minimum time in a mode before they can switch to another mode due to mechanical reasons, power constraints, information delays, stability considerations etc. This minimum time is known as the dwell time of the mode, a term previously used in the context of stability of switched linear systems [51], [41]. This important and practical constraint has not received attention in research on optimal control of switched systems and thus solving the general hybrid optimal control problem presented in the previous chapter under the dwell time constraints is the main focus of this work.

In this chapter, we consider the common optimal control problem of minimizing a cost functional defined on the state trajectory of nonlinear switched dynamical system under dwell time constraints where the control input consists of mode schedule i.e the continuous control input u is absent. The main contribution of the work presented in this chapter is the investigation of the topology of the optimization space under dwell time constraints and consequently, the need for replacing the notion of local optimality by stationarity with regards to an optimality function. Secondly, an optimality function is proposed to characterize the optimal solution. Finally, a

conceptual algorithm to solve the optimal mode scheduling problem based on this optimality function is presented and its convergence is proved.

While the gradient descent algorithms in [10], [36] and [76] are shown to be convergent, the question is what do they converge to, i.e. what is the local minima which leads to the question of what is the topology of the optimization space and whether these results can be extended to the case when the optimization space is restricted due to dwell time constraints. To answer this question, we explore in detail the topologies generated by the metrics used in [36] and [76] and investigate the impact of dwell time constraints on the structure of optimization space. We show that in the presence of dwell time constraints, the optimization space lacks structure and imposing a topology to define local minima that is useful and resonates well with the intuitive understanding of the problem is not possible. We propose to replace the notion of local optimality by stationarity and define the optimal solution for the scheduling problem as stationary points that satisfy optimality condition defined in terms of optimality function.

Next, we propose an optimality function for the mode scheduling problem with dwell time constraints. While the optimality functions in [10], [36], [76] use insertion gradients derived in [26] using variational principles, the presence of dwell time constraint makes such techniques from calculus ineffective since these methods insert modes at a point. This problem is addressed by defining our optimality function in terms of the cost differential resulting from mode insertion for duration equal to dwell time. In this chapter, we assume that each mode has the same minimum dwell time.

Lastly, we present a conceptual algorithm to solve the dwell time problem that uses a two step strategy similar to the ones in [10] and [36]. At the lower level, we solve the timing optimization problem with dwell time constraints using gradient projection method. At the higher level, we do a single mode insertion for duration of dwell time at the point that gives the maximum decrease in the cost while ensuring

that the dwell time constraints after mode insertion are still respected. The algorithm convergence is proved using the property of sufficient descent. We begin with formal problem definition of the dwell time problem.

3.1 Problem Formulation

We consider nonlinear switched dynamical systems in which there is no external input. The dynamics of such systems can be mathematically described by differential equations of the form

$$\dot{x}(t) = f_{v(t)}(x(t)), \tag{5}$$

where $\Phi = \{f_q \quad | \quad q \in \mathcal{Q}\}$ is a family of modal functions from R^n to R^n parameterized by finite index set $\mathcal{Q} = \{1, 2, \ldots, Q\}$ and $v : [0,\, t_f] \to \mathcal{Q}$ is a left continuous piecewise constant function of time referred to as the switching signal in chapter 2.

For schedule of length $n+1$, (5) can be written in the form

$$\dot{x}(t) = f_{\sigma(i)}(x(t)) \quad \forall \quad t \in (\tau_{i-1}, \tau_i] \tag{6}$$

and $i = 1, \ldots, n+1$. While (5) is more compact, we will use the representation in (6) in this chapter since it explicitly shows the mode schedule.

Suppose the minimum dwell time for each mode is some $\delta > 0$ and we are optimizing over finite horizon. Then the maximum number of modes that can exist in the interval $[0,\, t_f]$ is $N = \left\lfloor \frac{T}{\delta} \right\rfloor$ where $\lfloor \cdot \rfloor$ denotes the floor operator. A mode schedule acts as a feasible control input for our system if it satisfies the dwell time constraints

$$\Delta\tau_i = \tau_i - \tau_{i-1} \geq \delta, \tag{7}$$

for all $i = 1, \ldots n+1$. The collection of all such feasible mode schedules constitute the optimization space $\mathcal{X}$. With this in mind, we explicitly define our optimization space.

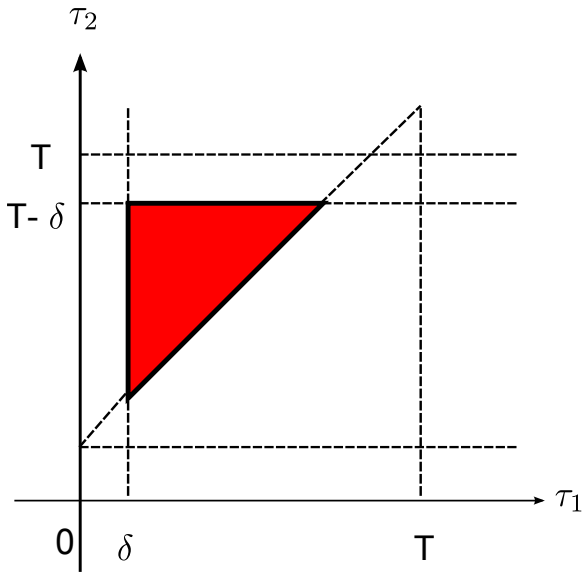


Figure 3: The switch time optimization space for the case of only two switches. The optimization region is shown in red.

For any positive integer n, the mode sequence σ is an element of $\Sigma^n = \mathcal{Q}^n$. Then the mode sequence space

$$\Sigma = \bigcup_{n=1}^{N} \Sigma^n,$$

is collection of all such mode sequences. The mode sequence space is a finite space.

Also for any positive integer n, the dwell time constraints (7) define a polyhedron

$$\mathcal{S}^n = \{\tau \in R^n \quad | \quad a_j \tau \geq b_j, \quad j = 1, 2, ..., n\}, \tag{8}$$

where $a_1 = (1, 0, 0, ..., 0)$, $b_1 = \delta, a_n = (0, ..., 0, -1)$, $b_n = \delta - T$ and for the rest $a_j = (0, ., -1, 1, ..0, 0)$, $b_j = \delta$ with the non zero entries of a_j at the j and $j + 1$ positions. $\mathcal{S}^n$ is a compact and convex subset of R^n. The transition time space is

$$\mathcal{S} = \bigcup_{n=1}^{N} \mathcal{S}^n.$$

The dwell time constraints automatically ensure that the transitions times τ_i are monotone increasing, i.e $\tau_{j-1} < \tau j$ for all $j = 1, \ldots, n + 1$.

For any positive integer n, let

$$\mathcal{X}^n = \Sigma^n \times \mathcal{S}^{n-1}.$$

So the mode schedule $\xi \in \mathcal{X}^n$ is the tuple $\xi = (\sigma, \tau)$. The optimization space, which is the collection of all such mode schedules is then

$$\mathcal{X} = \bigcup_{n=1}^{N} \mathcal{X}^n. \tag{9}$$

This optimization space is finite dimensional and is a subset of the infinite dimensional optimization space that would result if no dwell time constraints were considered.

Dwell Time Problem

Let $L : R^n \to R$ be a cost function defined on the state trajectory $x(t)$, and consider the cost functional $J : \mathcal{X} \to R$, defined by

$$J(\xi) = \int_0^{t_f} L(x(t))dt + \phi(x(t_f)). \tag{10}$$

Let $x_0 \in R^n$ be the initial condition of the system (6). The dwell time problem is to solve

$$\min_{\xi \in \mathcal{X}} \quad J(\xi), \tag{11}$$

subject to the dynamical constraints (6) and initial condition constraints . To ensure that a unique bounded solution exists for the differential equation, we make the following mild assumptions which are fairly standard in switched optimal control problems [25], [10].

Assumptions:

(i) The modal functions $f \in \Phi$ and cost function L are continuously differentiable.

(ii) There exists a constant $C > 0$ such that for every $x \in R^n$ and every $f \in \Phi$ we have $||f(x)|| \leq C(||x|| + 1)$.

3.2 Topology and Local Minima

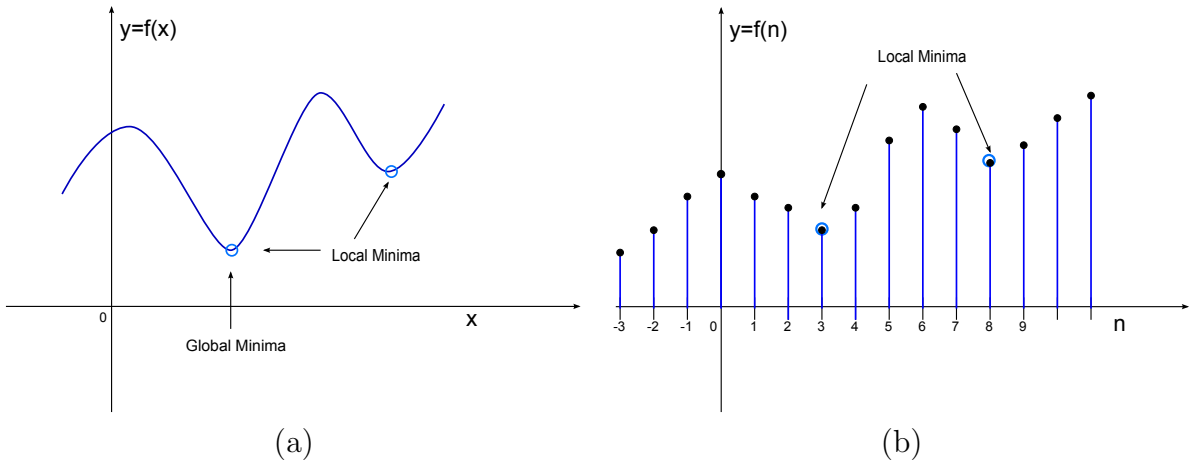
In this section, we begin with an explanation of why we define the minima for the dwell time problem using optimality functions and why defining local minima for

such problems is not useful. The definition of local minima for the optimization space requires the concept of neighborhood which in turn depends on the topology defined on the space. While many topologies can be defined for the same optimization space, the choice of topology should be such that it resonates well with the underlying problem. Before we proceed with understanding the structure of optimization space, we state what is meant by a neighborhood in a given topological space.

Definition 1 *Given a topological space (X, T), a subset N_x of X is a neighborhood of a point $x \in X$ if N_x contains an open set $U \subset T$ containing the point x.*

We give a couple of examples that will prove useful in the discussion below. Let $X = \{a, b, c\}$. For the trivial topology $T = \{\emptyset, \{a, b, c\}\}$, the points in the search space cannot be distinguished by topological means and either all or no points are neighbors toe each other. For the topology $T = \{\emptyset, \{a\}, \{a, b\}, \{a, b, c\}\}$, the points a and b are neighbors.

To motivate what is to follow about the topology on our spaces, we consider few scenarios concerning the open neighborhoods and local minima. Suppose we have a continuous function $f : R \to R$ and we want to find the point where the function is locally minimum. Then the decision variable in this case is continuous in the sense it takes values in R and we call a point $x_0 \in R$ as its local minimum point if we can find an $\epsilon > 0$ ball such that $f(x_0) \leq f(x)$ for all x in this ball. The ϵ ball defines a neighborhood of point x_0 in this case. Suppose now the decision variable is discrete, for example, the set of integers and let say $f : Z \to R$. The ϵ ball definition of local minima does not make sense in this case since for $0 < \epsilon < 1$ the ball will contain only the point itself thereby making every point as a local minima. A more reasonable way of declaring a point n as a local minima would be if $f(n)$ is less than $f(n-1)$ and $f(n+1)$. The points $\{n-1, n+1\}$ constitute the neighborhood of point n. The point highlighted in this paragraph is shown in Fig. 4



(a) (b)

Figure 4: Local minima for continuous functions and discrete functions. The usual topology of R^n works for continuous functions but needs extension for discrete case.

In case of the transition time space, the cost functional J is a continuous function of the decision variable $\tau \in R^n$. Hence the choice for topology is the usual topology of R^n obtained by the cartesian product of open intervals in R. However the mode sequence space is a discrete space and one would think that a reasonable definition of local minima for function defined on this space would be some natural extension of the local minima defined above for a discrete space. The problem is that while in the case of integers for example, it was easy to see that the points $n-1$ and $n+1$ are the neighbors of point n, we don't have such an obvious extension in the case of mode sequence space. To address this problem, one would like to define a metric on the space which can tell us which points are closer to a point than others. The questions is does such a metric exists and is it useful. We first look at this question without dwell time constraints and then in the presence of dwell time constraints.

In [36], a metric is defined on the mode sequence space in a form inspired by the notion of Hamming distance between vectors and a metric on transition time space is defined using ℓ_1 norm. The external input considered in [36] is ignored here without affecting our discussion. For the optimization space $\mathcal{X}$, which is a product space of mode sequence space Σ and transition time space $\mathcal{S}$, the metric is defined as the sum

of the metrics on the transition time space and mode sequence space as

$$d(\xi_x, \xi_y) = \{\sigma_x \neq \sigma_y\}\vec{1} + ||\tau_x - \tau_y||_1, \tag{12}$$

and the algorithm convergence is proved using the property of sequential continuity in metric spaces. The topology generated by this metric is such that the ϵ neighborhood of a point ξ_0 will contain only points of transition time space and no points from the mode sequence space.

In [78], the mode schedules are represented using switching functions $v(t)$ corresponding to mode schedules ξ and the topology is induced on the optimization space $\mathcal{V}$ using the L_1 norm

$$d(v_x, v_y) = ||v_x - v_y||_{L_1}. \tag{13}$$

In the topology generated by this metric, the ϵ neighborhood of a point ξ_0 contains points of mode schedule space as well as transition time space when considered in the mode schedule representation form. When two switching functions are closed to each other in the L_1 norm, the corresponding state and costate trajectories are close to each other in the L_∞ norm. The cost function J is thus a continuous function of $v(t)$ and a definition of local minima is obtained that resonates well the problem.

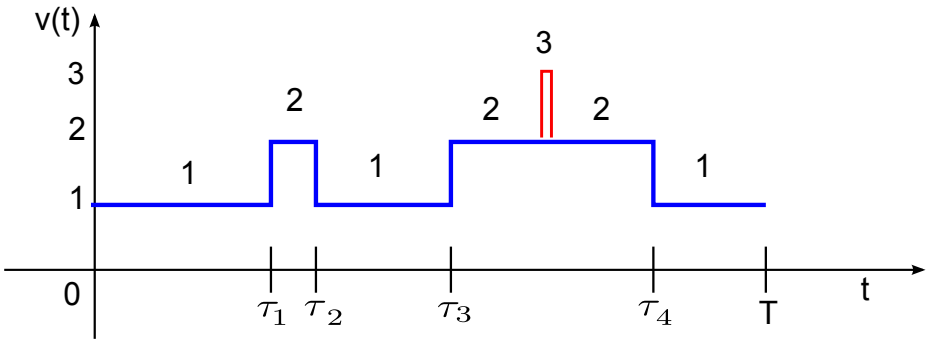


Fig: Switching Function

Figure 5: Modifying the switching signal by inserting a new mode for duration of *epsilon*. The corresponding trajectories are close in L_∞ norm whenever the switching functions are close to each other in L_1 norm.

The scenario changes when dwell time constraints are imposed on the optimization space. The ϵ neighborhoods generated by the metric in [78] or [36] under dwell time constraints do not contain mode sequence points but just transition space points. If we use Hamming metric on mode sequence space and use open balls of radius 2 using metric in (12) to define neighborhoods that contain mode sequence points, we still run into problems. To see the neighborhoods generated as a result of this metric, consider the case where $\mathcal{N} = \{1, 2, 3\}$ and $N = 2$ due to dwell time constraints. Then the smallest non-trivial neighborhood of mode sequence point $(1, 2)$ include the sequence points $(1, 3)$ and $(3, 2)$ both of which are distance 1 apart from the point under consideration. The point $(1, 2)$ will be then locally optimal w.r.t. cost functional J if the cost associated with this mode sequence is less than its two neighboring points which constitute its neighborhood. The sequence point $(3, 2)$ is distance 2 apart from point $(1, 2)$ in the metric.

While the definitions are consistent, this does not resonate well with the problem we are trying to solve. To see this, even for fixed switching times, the state trajectory $x(t)$ resulting from mode sequence $(3, 2)$ might be closer to the optimal state trajectory resulting from $(1, 2)$ than the mode sequences $(1, 3)$ and $(3, 2)$ in L^p norm which were in its neighborhood and thus does not appeal to our intuitive understanding of the local minima for the actual problem. The same problem results when we consider the metric (13) that uses switching functions.

The problem primarily lies not in the metric used in [36] or (13) in the sense that any other meaningful metric defined on this space would result in similar difficulties. The problem lies in the inherent lack of structure in the optimization space due to dwell time constraints. There exists no meaningful similarities between different elements of the optimization space. Using metric in problems such as this which lacks meaningful similarities between different decision alternatives is problematic since it creates similarities where no similarities exist. In such cases the feasible topology is

the trivial topology which that all points are neighbors to any point [63].

To summarize the main point of our discussion, while a useful notion of local minima exists for the hybrid optimal control problems in the absence of dwell time constraints using the topology induced by L_1 norm on the switching function space, we cannot have a similar notion when dwell time constraints are present. Instead of defining a joint topology on the transition time space and mode sequence space for defining local minima, we then say a point $\xi_0 = (\sigma_0, \tau_0) \in \mathcal{X}$ is a local minimum if τ_0 is a local minimum in the topology induced by R^n and σ_0 is a local minimum in the topology induced by the hamming metric on the mode sequence space.

3.3 Optimality Functions and Optimality Conditions

To define the minima for our dwell time problem, we resort to the use of optimality functions which are semicontinuous functions of the form $\theta : X \to R^-$. Points in X that satisfy the optimality condition $\theta_X(x) = 0$ are referred to as stationary points and constitute the optimal solutions for our dwell time problem. The magnitude of optimality function $|\theta_X(\cdot)|$ can be seen as a measure by which point $\xi \in \mathcal{X}$ fails to satisfy the optimality condition. Next we propose the optimality function for the dwell time problem.

The optimality functions used in [10], [36] and [76] rely on insertion gradient which tests the sensitivity of cost due to each mode over an arbitrarily small interval. Due to the presence of dwell time constraints, such calculus based gradient insertion at a point does not work for us. We propose an optimality function on the cost differential resulting from a feasible mode insertion for duration of minimum dwell time.

Definition 2 *A mode insertion is feasible if the resulting mode sequence does not violate the dwell time constraints.*

Let $U \subset [0, T]$ be set of all points where the mode insertion is feasible. At given iteration k, let σ_0 be time optimized mode sequence to which we want to insert a new

mode. Let J_{k0} be the cost associated with this mode sequence. Let $J(t, f_j)$ denote the cost associated with mode sequence obtained by replacing the mode at time $t \in U$ by mode f_j for δ seconds. Define

$$J_\Delta(t, f_j) = J(t, f_j) - J_{k0}. \tag{14}$$

We then define our optimality function as

$$\theta_{\mathcal{X}} = \min \{ J_\Delta(t, f) \quad | \quad f \in \Phi, \quad t \in U \}. \tag{15}$$

Now $\theta_{\mathcal{X}} \leq 0$ since we can always insert the same mode. The condition $\theta_{\mathcal{X}} = 0$ is then the necessary condition for optimality which we refer to as the optimality condition. We define the optimal solution for the dwell time problem as follows

Definition 3 *A point $\xi_0 = (\sigma_0, \tau_0) \in \mathcal{X}$ is optimal solution for the dwell time problem (11) if τ_0 is Karush Kuhn Tucker (KKT) point for the timing optimization problem and $\theta_{\mathcal{X}}(\xi_0) = 0$.*

Computing analytical solutions to optimal control problem is hard and thus one needs to devise a strategy for computing them. Next, we first present a conceptual algorithm for computing the optimal solutions and then next present an efficient method for arriving at the same results in the next chapter.

3.4 Algorithm - Conceptual

In this section we present a conceptual algorithm that employs a basic two step strategy similar to the one used in [10] and [36]. At the higher level, the mode sequence is optimized by inserting a single mode for δ seconds and at the lower level, for a fixed mode sequence, the switching times between different modes are optimized subject to the dwell time constraints.

- Perform Switch Time Optimization.

- Perform Mode Sequence Optimization.
- Check for Optimality Conditions.

We begin with the switch time optimization problem first.

3.4.1 Switch Time Optimization

For a fixed mode sequence σ_0 of length $n+1$, the cost functional in (137) is only function of $\tau \in \mathcal{S}^n$ and the switch time optimization problem becomes

$$\min_{\tau \in \mathcal{S}^n} J(\tau) \tag{16}$$

subject to 6 and initial condition constraints. In [26], the derivative of the cost with respect to switching times for this problem without dwell time constraints is shown to be

$$\frac{dJ}{d\tau_i} = p(\tau_i)\left(f_{\sigma_0(i)}(x(\tau_i)) - f_{\sigma_0(i+1)}(x(\tau_i))\right) \tag{17}$$

where $x(t)$ is the solution of the system (6) and $p(t)$ is solution of the costate equation

$$\dot{p}(t) = -p(t)\frac{\partial f_{\sigma_0}(x(t))}{\partial x} - \frac{\partial L(x(t))}{\partial x}, \quad p(t_f) = \frac{\partial \phi(x(t_f))}{\partial x}. \tag{18}$$

If $\nabla J(\tau)$ denotes the gradient of the cost J with respect to the switching time vector τ, the timing optimization problem is then solved using steepest descent algorithm with Armijo step size by stepping in direction opposite to gradient.

For sets that are convex and compact, we can find the feasible descent direction by taking the projection of vector found using steepest descent algorithm onto the constraint set, which in general requires solving a quadratic optimization problem. When the constraint set has the structure of polyhedron, the direction finding problem can be greatly simplified using manifold suboptimization methods based on [62] which are a type of gradient projection methods. In this method, instead of projection onto the entire constraint set, the gradient is projected on a linear manifold of active

constraints which makes computation of the projection quite easy [13]. Since our constraint set $\mathcal{S}^n$ has the structure of a polyhedron, we use this method to solve the timing optimization problem.

Let $\mathcal{I}(\tau_k)$ denote the index set corresponding to active constraints at feasible point τ_k, i.e.

$$\mathcal{I}(\tau_k) = \{j \quad | \quad a_j\tau_k = b_j, \quad j = 1, 2, .., r\}$$

and we assume without loss of generality that vectors $\{a_j, j \in \mathcal{I}(\tau_k)\}$ are linearly independent at every τ_k. The feasible descent direction is obtained from the subspace

$$M(\tau_k) = \{d \quad | \quad a_j h_k = b_j, \quad j \in A(\tau)\}.$$

This is the subspace which is parallel to the manifold of active constraints. The projection of the gradient onto $M(\tau_k)$ can be obtained by solving the following quadratic optimization problem.

$$\min_{h_k \in M(\tau_k)} \nabla J(\tau_k)' h_k + \frac{1}{2} h_k' h_k.$$

The unique optimal solution of this problem can be easily computed from Karush Kuhn Tucker (KKT) conditions to be

$$h_k = -P_M \nabla J(\tau_k), \tag{19}$$

where

$$P_M = I - L_k^{'}(L_k L_k^{'})^{-1} L_k, \tag{20}$$

is the projection matrix and L_k is the matrix that has as rows the vectors $a_j, j \in \mathcal{I}(\tau_k)$. The KKT multiplier μ_k for this problem is given by

$$\mu_k = -(L_k L_k^{'})^{-1} L_k^{'} \nabla J(\tau_k). \tag{21}$$

If no constraint is active, then we have $P_M = I$ from (20) and the feasible descent direction is $h_k = -\nabla J(\tau_k)$ from (78), the usual steepest descent direction.

To ensure that the algorithm converges, the step size in the descent direction h_k is obtained by using Armijo step size rule [9] over the set

$$\Lambda = \{\lambda > 0 \quad | \quad a_j'(\tau_k + \lambda h_k) \geq b_j, \quad j \notin \mathcal{I}(\tau_k)\}.$$

From this we can compute the upper limit on λ_k as

$$\lambda_{max} = \min_{j \notin \mathcal{I}(\tau_k)} \frac{a_j \tau_k - b_j}{a_j h_k}, \tag{22}$$

such that $a_j h_k < 0$, otherwise there is no upper limit due to the j^{th} inactive constraint on λ. In case there is no upper limit on λ, we set $\lambda_{max} = \bar{\lambda}$ for some fixed $\bar{\lambda}$. So we have $\lambda_k = c_k \lambda_{max}$ where $c_k \in (0, 1]$. Let $h_k' = \lambda_k h_k$ be the scaled descent direction then the next iteration can be written as

$$\tau_{k+1} = \tau_k + c_k h_k'. \tag{23}$$

Let $\alpha \in (0, 1]$, $\beta \in (0, 1)$ and let m be the smallest positive integer for which

$$J(\tau + \beta^m h_k') - J(\tau_k) \leq \beta^m \alpha \langle \nabla J(\tau_k), h_k' \rangle. \tag{24}$$

Then $c_k = \beta^m$ is our step size to be used for updating τ_{k+1} via (82).

When $h_k = 0$, the algorithm can be terminated if all components of μ_k are non-negative, indicating that the Kuhn Tucker conditions are satisfied. If some components of μ_k are negative, it is possible to make progress by removing from active constraint associated with it. In general, the constraint associated with the most negative constraint is removed and the descent direction h_k is computed again. If $h_k \neq 0$, we can proceed as usual to compute the step size and update τ_k. If $h_k = 0$ and there are still negative components of μ_k, we keep on dropping the corresponding active constraints until all components of μ_k become non-negative. The switch time optimization algorithm can be summarized as follows.

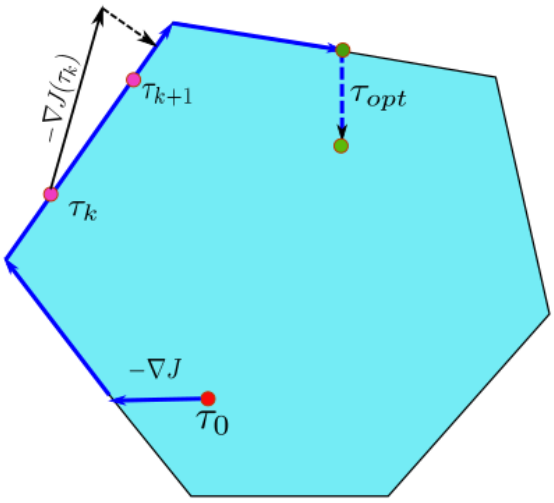


Figure 6: Manifold Suboptimization. The descent direction is taken from the set of active constraints. The gradient is projected onto the set of active constraints. The constraint is dropped if the corresponding Lagrange multiplier is negative at a sub-optimal point.

ALGORITHM - STO

Given a fixed mode sequence σ_0 and a vector τ_k containing the switching times between different modes, identify the set of active constraints $\mathcal{I}(\tau_k)$, choose $\bar{\lambda}$, set $k = 0$ and do the following.

1. Form matrix L_k corresponding to active constraints and compute the projection matrix P_M using (20).

2. Compute the cost gradient $\nabla J(\tau_k)$ via (93) and the descent direction h_k using (78).

3. If $h_k = 0$, compute vector μ_k using (80). If all entries of μ_k are non-negative **stop**. If not, eliminate the active constraint associated with the most negative μ_k and go to step 1.

4. Compute the maximum step size λ using (22) and step size c_k using Armijo Rule via (83).

5. Update the τ using (82) and the set of active constraints $\mathcal{I}(\tau_k)$. Set $k = k + 1$ and go to step 1.

The Armijo step size ensures that the switch time optimization algorithm *STO* has the property of sufficient descent, which will be explained in the next section, and hence the algorithm converges to stationary points of the optimal control problem. We next demonstrate the application of our algorithm by considering its application to an example problem.

3.4.1.1 Example

We consider the linear system in [26] that can switch between two modes and which we will consider throughout this thesis for example purposes and we will refer to as *'Two Switched System'*. The matrices associated with the dynamics of two linear subsystems are

$$A_1 = \begin{pmatrix} -1 & 0 \\ 1 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 1 \\ 1 & -2 \end{pmatrix}.$$

We define a quadratic cost functional on the state trajectory.

$$J = \frac{1}{2} \int_0^T ||x(t)||^2 dt.$$

Suppose the dwell time constraint associated with both modes having a minimum dwell time of $\delta = 0.2$s. We consider the sequence $\sigma_0 = \{1, 2, 1, 2, 1\}$. We start with a feasible τ and solve the problem over the horizon $t_f = 2$ seconds. The Armijo parameters are set to $\alpha = 0.01$ and $\beta = 0.5$ and the differential equations are solved using a step size of $dt = 0.0001$ s. The results are shown in Fig. 7 - 11 and we next briefly explain them.

Fig. 7 shows the cost vs. iterations plot. The algorithm converges in about 131 iterations and the cost reduces from the initial value of 10.81 to the final value of 2.60. The sudden decrease in cost at some stages occurs as some of the constraints

are dropped during the optimization process corresponding to negative Lagrange multipliers. The staircase appearance indicates rapid convergence at each manifold sub-optimization step, thanks to Armijo step size rule. Fig. 8 shows the behavior of the cost from iteration 15 onwards, to highlight the submanifold optimization occurring during the switch time optimization process

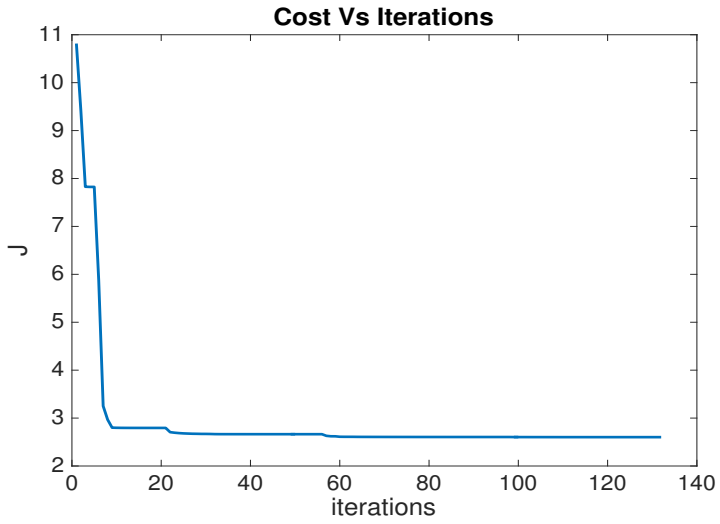


Figure 7: Plot of cost vs. iterations. Every time a constraint is dropped, there is a rapid decrease in cost. The process repeats till convergence is achieved.

The optimal switching times between different modes subject to the above mentioned constraints are shown in Fig. 9. The switching times, which were initialized to $\tau = (0, 0.2, 0.5, 0.9, 1.1, 2)^\top$, are optimized to $\tau_{opt} = (0, 0.67, 1.06, 1.34, 1.77, 2)^\top$. Fig. 9 shows the transitions between the two modes at the end of optimization.

Fig. 10 shows the evolution o optimal state trajectory with time. Fig. 11 shows the insertion gradients at the end of switch time optimization. As expected, the insertion gradients at the optimal switching times for the inactive constraints are zero.

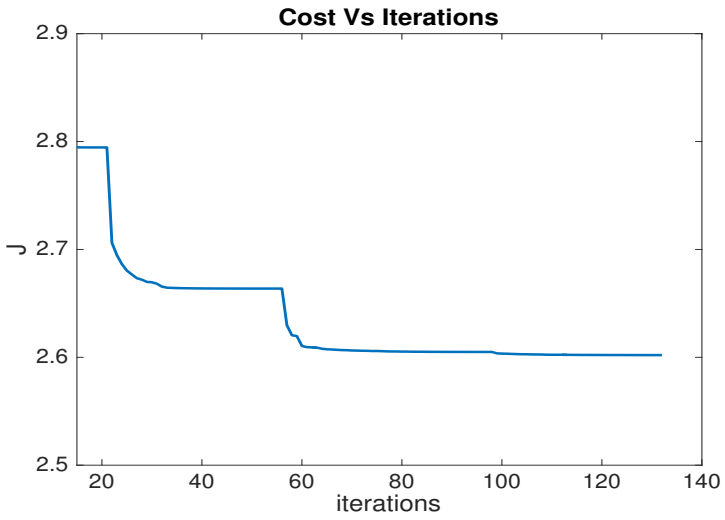


Figure 8: Tail of the cost in the above figure showing the behavior after iteration 15 and showing the same sudden decrease in cost as the constraints are dropped during the manifold suboptimization process.

3.4.2 Mode Sequence Optimization - Conceptual

There is one more issue that needs to be addressed to start feasible. Since the mode inserted has to exist at least for δ seconds, we cannot come very close to the existing switching times, otherwise there dwell time constraint will be violated. In other words, there is a $U \subset [0, T]$ where the mode insertion is feasible. This problem is addressed by searching in the areas at and between switching intervals shown bold in the Fig. 12, assuming that the difference $\tau_{k+1} - \tau_k \geq 3\delta$. In this case, we check for insertion at points $P_{k+1} = \{\tau_k, \tau_{k+1} - \delta\}$ and the interval $I_{k+1} = [\tau_k + \delta, \tau_{k+1} - 2\delta]$. For the case where $2\delta \leq \tau_{k+1} - \tau_k < 3\delta$, the only points to be checked for insertion are $P_{k+1} = \{\tau_k, \tau_{k+1} - \delta\}$. If $\delta < \tau_{k+1} - \tau_k < 2\delta$, we do not insert any mode. If $\tau_{k+1} - \tau_k = \delta$, then $P_{k+1} = \{\tau_k\}$ is the only point to be checked for insertion. For mode schedule ξ of length n, the feasible insertion region is then

$$U = \bigcup_{k=1}^{n} (P_k \cup I_k).$$

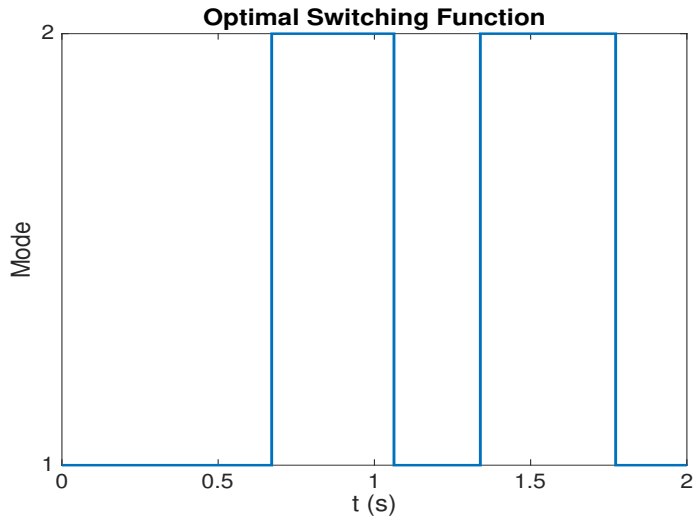


Figure 9: Optimal transition times between the two modes after performing switch time optimization subject to dwell time constraints.

At given iteration k, consider the cost differential (14) associated with time optimized mode sequence σ_0 to which we want to insert a new mode.

$$J_\Delta(t, f_j) = J(t, f_j) - J_{k0}. \tag{25}$$

The mode to be inserted and its location is given by argmin of the optimality function $\theta_\mathcal{X}$ defined in (15) and can be determined as follows. For every $k \in \mathcal{N}$ compute

$$t_k^* = \arg\min_{t \in U} J_\Delta(t, f_j), \tag{26}$$

$$J_\Delta(t_k^*, f_k) = J(t_k^*, f_k) - J_{k0}. \tag{27}$$

Then the insertion point and the mode to be inserted are given by

$$(t^*, f^*) = \arg\min_{k \in \mathcal{N}} J_\Delta(t_k^*, f_k), \tag{28}$$

so long as $J_\Delta(t_k^*, f_k) < 0$.

Solving (26) to find the optimal insertion point for every mode is computationally expensive since it seeks minimum over the entire set $U \subset [0, T]$. As opposed to the optimal insertion point in [26] which utilizes the one time computed state and costate

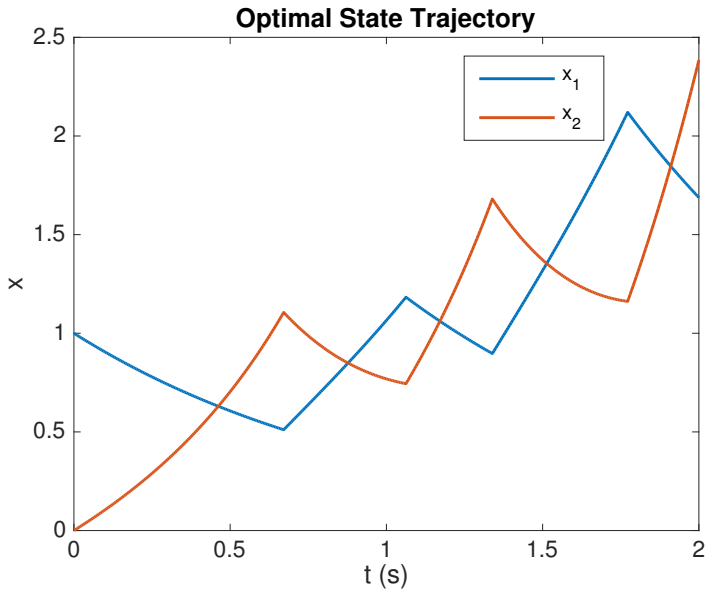


Figure 10: Plot shows the evolution of state trajectory as a function of time. The transitions correspond to the switching between different modes.

trajectory, this requires solving the state and costate trajectory for each point to be checked for insertion and we will address this problem in the next chapter.

At each iteration, we identify the point of insertion t^* and the mode to be inserted f^* via (26) - (28). After this we use the projected gradient algorithm to optimize the switching times subject to the dwell time constraints. The process is repeated until no reduction in cost can be achieved by inserting a new mode. Thus $\theta_{\mathcal{X}} = 0$ is the optimality condition and the corresponding mode sequence σ^* is the optimal solution for the mode sequencing problem and τ^* is the solution for switch time optimization problem. Taken together $\xi^* = (\sigma^*, \tau^*)$ is the optimal solution for the dwell time problem. We summarize the algorithm below.

Dwell Time Algorithm - I

Given A mode sequence σ having n modes and vector τ satisfying the dwell time constraints and time optimized, repeat the following three steps:

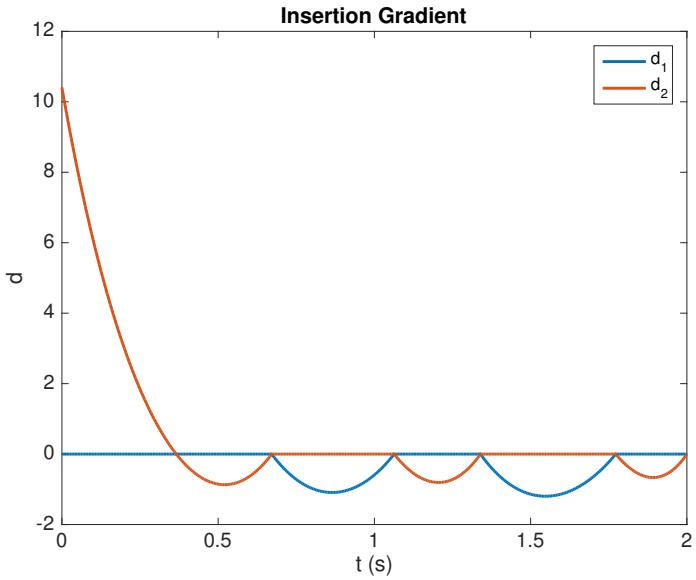


Figure 11: Insertion gradients for the two modes. When no constraints are active, the insertion gradients are zero at the transition points.

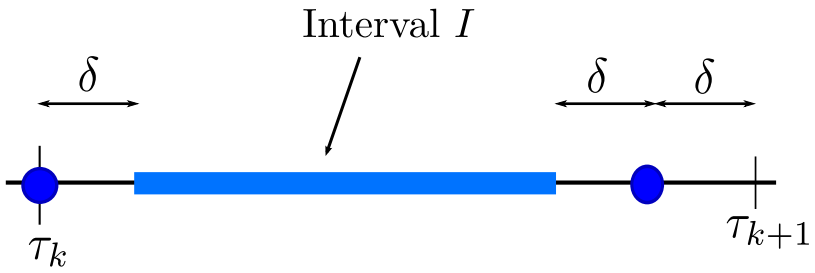


Figure 12: Scanning area for mode insertion. The interval I and the points marked as circles form region of feasible mode insertion.

1. Compute the optimality function $\theta_{\mathcal{X}}$ as defined in (15).

2. If $\theta_{\mathcal{X}} = 0$ then *stop*. Otherwise insert mode f^* in the interval $[t^*, t^* + \delta]$, by appending two new switching instants to vector τ at times t^* and $t^* + \delta$. We get new mode sequence σ and switching time vector τ.

3. For new σ, solve for optimal τ using gradient projection algorithm.

3.4.3 Convergence

To prove the convergence of our algorithm, we follow the approach used in [57] that relies on the property of sufficient descent. Let $\{\xi_j\}$ be the sequence of points computed by the algorithm, then the property of sufficient descent for an algorithm as defined as follows.

Definition 4 *An algorithm $a : \mathcal{X} \to \mathcal{X}$ has the property of sufficient descent with respect to an optimality function θ if for every $\delta' > 0$, there exists $\eta > 0$ such that when $\theta(\xi_j) < -\delta'$ then*

$$J(\xi_{j+1}) - J(\xi_j) < -\eta. \tag{29}$$

Since $J(\xi) \geq 0$ for all $\xi \in \mathcal{X}$, the algorithm having the property of sufficient descent converges i.e for every infinite sequence $\{\xi_j\}$ computed by algorithm, $\lim_{j\to\infty} \theta(\xi_j) = 0$. Otherwise $\lim_{j\to\infty} J(\xi_j) = -\infty$ which contradicts the fact that J is bounded from below. The convergence is obvious if $\{\xi_j\}$ is a finite sequence.

Lemma 1 *The dwell time algorithm has the property of sufficient descent.*

Proof 1 *Consider our optimality function (15). Since $\theta_{\mathcal{X}}(\xi_j) = J(\xi_{j+1}) - J(\xi_j)$ and $\theta_{\mathcal{X}}(\xi_j) < -\delta'$ with $\delta' > 0$ at all non-stationary points, we have $J(\xi_{j+1}) - J(\xi_j) < -\eta$ with $\eta = \delta'$. Hence proved.*

Again, since our optimization space is finite dimensional, we do not run into problems associated with infinite dimensional parameter spaces [10] to prove the sufficient descent property of our algorithm. We now state the main theorem about algorithm's convergence.

Theorem 1 *The dwell time algorithm converges to stationary points of the optimality function $\theta_{\mathcal{X}}$.*

Proof 2 *The proof follows immediately from the property of sufficient descent proved for our algorithm and the discussion following the definition of sufficient descent regarding algorithm convergence.*

.

3.5 Example

We consider the same *two switched system* with quadratic cost example as before. The switched system is initialized to $x_0 = (1, 0)'$ and optimized over time interval $[0, 2]$. The Armijo parameters are set to $\alpha = 0.5$ and $\beta = 0.5$. The time step for simulation was set to $dt = 1e^{-3}$ s. For intervals in the set $U \subset [0, T]$ for feasible mode insertion, the time step used was $ds = 1e^{-3}$ s. The mode sequence is initialized to $\sigma_0 = \{f_1\}$ for which the cost is $J = 40.75$. The algorithm is terminated when $|\theta_{\mathcal{X}}(\cdot)|$ falls below $\epsilon = 0.1e^{-3}$. The simulation is performed for dwell time constraint of $\delta = 0.01$s for both modes and results are shown in Fig. 13-16. To emphasize the variation of cost with each iteration, the initial cost $J = 40.75$ which is relatively large is not shown in Fig. 13 and the graph starts with iteration 1. It takes about 7 iterations before the optimality conditions are met and the final cost is $J = 2.40$.

The cost differentials resulting from the sweeps of both modes at the optimal solution are shown in Fig. 14, indicating that introducing either of the modes in the feasible region would not result in any decrease in cost and hence the optimality conditions are satisfied. The optimal switching function is shown in Fig. 15 while the evolution of state trajectory with time is shown in Fig. 16. For the case $\delta = 0.1$ s, it takes about 3 iterations before the optimality conditions are satisfied. The final cost $J = 2.60$ is relatively higher than with smaller dwell time constraints as expected. Also due to the presence of larger dwell time constraints in the second case, some of the modes that appear in the optimal mode schedule of Fig. 15 do not appear in optimal mode schedule associated with $\delta = 0.1$ s.

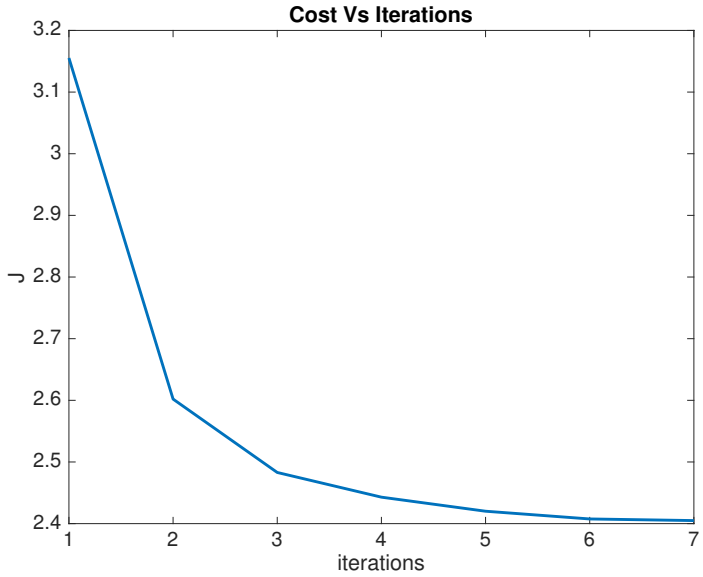


Figure 13: Plot of cost vs. iterations. The initial cost due to its large value is not shown to highlight the variation in cost.

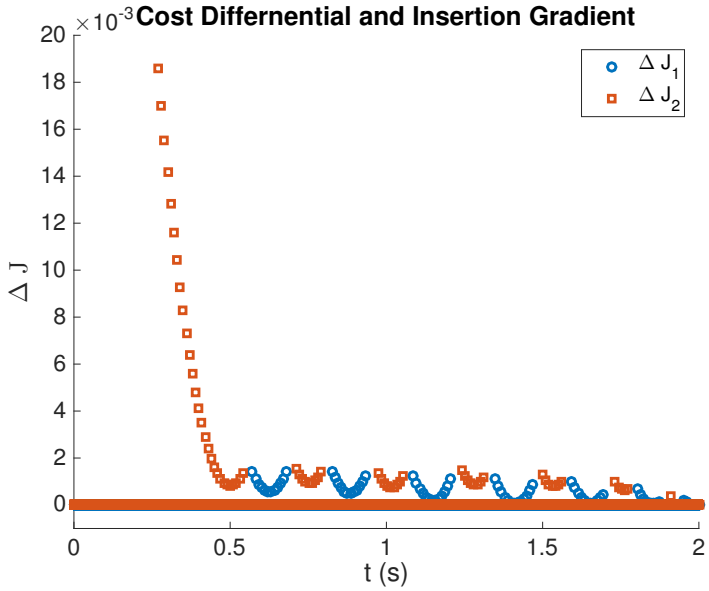


Figure 14: The change in cost due to insertion of both modes. At the optimal solution, both the trajectories are positive, indicating inserting a new mode anywhere will result in an increase in cost. The mode is inserted only in feasible insertion regions for duration of dwell times.

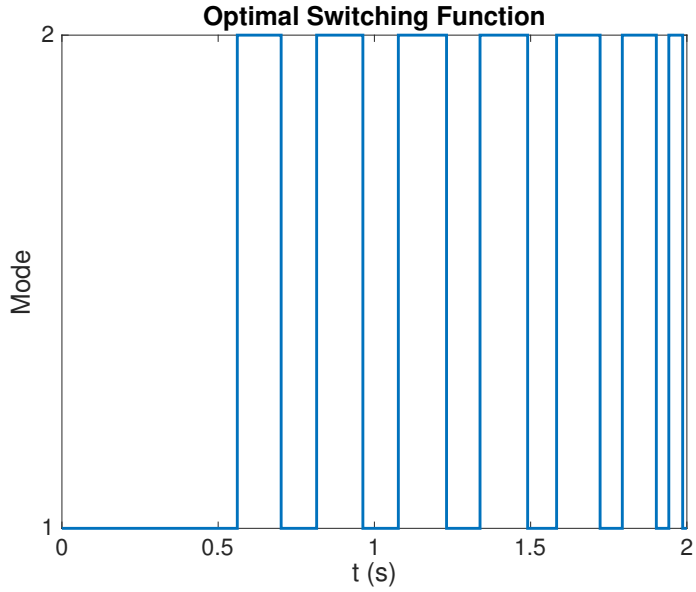


Figure 15: Optimal Switching function. The mode 1 exists initially for larger duration to bring down the component of state x_1 having large value.

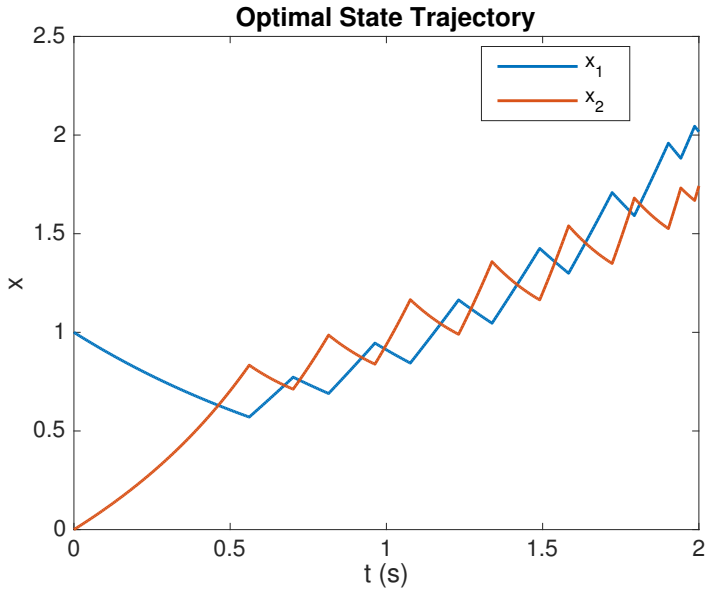


Figure 16: Optimal state trajectory as a function of time. The abrupt transitions correspond to the switching between different modes.

The above procedure for computing the optimal schedule is kind of brute force manner for updating the mode sequence, searching at every point and a more efficient way of updating the mode sequence is thus desired to make our dwell time algorithm implementable and hence practically useful for solving complex problems and this is the topic of the next chapter.

CHAPTER IV

COMPUTING THE OPTIMAL SOLUTION FOR THE DWELL TIME PROBLEM

In the previous chapter, we presented a conceptual algorithm to compute the optimal control that employed a two step strategy by splitting the problem into timing optimization problem and mode sequence optimization subproblems. At the lower level, the algorithm optimizes the switch time optimization and at the higher level, the algorithm optimizes the mode sequence by inserting a mode for the duration of dwell time that results in a maximum decrease in the cost. While the switch time optimization part was solved using an effective gradient projection method, the mode sequence optimization was solved in a brute force fashion to find the location of the optimal mode insertion. This involves the computation of state and costate trajectory at every time step, which makes it impractical even for slightly complicated problems. For the dwell time algorithm presented in the previous chapter to be of any practical significance, and given that we can already handle the switch time optimization under dwell time constraints as detailed in the previous chapter, we need an efficient way for updating the mode sequence, and this will be the main focus of this chapter.

The main problem with the mode sequence optimization in [5] stems from the fact that the new mode has to exist at least for the duration of mode's dwell time and this makes the direct application of methods from variational calculus or needle variations such as the ones used in [10, 25, 78] inadequate for updating the mode sequence. In these methods, the mode insertion gradient, which in essence is the change in cost due to to insertion of a mode to the existing sequence at a point for arbitrarily small time, is computed for each mode at every point in time for updating

the mode sequence. Since the insertion is done at a point, the state and costate trajectory remains the same due to the integral effect and thus the computation of mode insertion gradient requires the pre-computed state and costate trajectory from the switch time optimization, which allows for fast computation and is a mainstay for the success of these methods. A technique, that makes use of the insertion gradient for the dwell time problem is thus highly desirable from the computational standpoint and arriving at one is one of the main focus of this chapter.

In [77], the authors attempted to solve this issue by utilizing the information of mode insertion gradient by integrating it over the duration of mode's dwell time for computing the change in cost. This can however only work for very small dwell times since when the mode is not inserted at a point but over a duration, the state and costate trajectories, from which the mode insertion gradient is computed changes and hence utilizing the insertion gradient based on pre-computed state and costate trajectories is no longer valid.

In this chapter, we dig deep into developing an understanding of the mode insertion gradient, in particular from the geometric viewpoint and from there move onto presenting a computational technique based on mode insertion gradients for updating the mode sequences and hence solving the mode scheduling problem for switched dynamical systems under dwell time constraints. We present the proof for the convergence of our algorithm in the framework of our proposed optimality function for the dwell time problem in the previous chapter. Next we consider the application of our method to optimal control problems for linear and nonlinear switched systems and compare the results with those from the brute force method in the previous chapter.

4.1 Geometric Properties of Mode Insertion Gradient

In this section, we explore the geometric properties of mode insertion gradient and make connections with existing results in optimal control to enable a deep understanding of this important quantity in the optimal control of hybrid dynamical systems. The analytical expression for mode insertion gradient due to the insertion of mode at any time $\tau \in [0\,, t_f]$ was derived in [25, 26] using calculus of variations by inserting a mode to the existing sequence at time τ for duration of time that in the limit goes to zero. The insertion gradient due to insertion of mode f_a to the existing mode sequence $f_{\sigma(t)}$ is given by

$$\frac{dJ}{d\tau}(f_a) = p(\tau)^\top \left(f_a(x(\tau)) - f_{\sigma(\tau)}(x(\tau))\right) \tag{30}$$

where $x(t)$ and $p(t)$ are the state and costate trajectories associated with the existing mode sequence σ. From the theory of optimal control [17], we know that the gradient of cost with respect to initial condition for the Bolza type problem is given by

$$\frac{dJ}{dx_{t_0}} = -p(t_0) \tag{31}$$

where $p(t)$ is the costate solution given by the solution of adjoint equation. This easily extends to the case, where the initial condition is not the point at time t_0 but any point along the state trajectory at some time τ in the interval $[t_0,\, t_f]$. This is because the trajectory, and hence the cost, in the interval $[t_0,\, \tau)$ remains unchanged due to this perturbation. The above expression for the gradient w.r.t perturbation in state trajectory at time τ then becomes

$$\frac{dJ}{dx_\tau} = -p(\tau) \tag{32}$$

where $x_\tau = x(\tau)$ is the state value being perturbed, and $p(\tau)$ is the costate value at time τ. Since the the gradient points in the direction of increase in cost, this equation tell us that if the initial condition $x(t_0)$ is perturbed in the direction of the costate

at t_0, i.e. $p(t_0)$, the cost will decrease and vice versa. Thus if the perturbation in the state $x(\tau)$ due to the insertion of mode f_a at any time τ is in a direction opposite to the vector $-p(\tau)$, the cost should decrease and vice versa. So how does the state $x(\tau)$ change due to the insertion of mode at single point τ? If $f_{\sigma(\tau)}$ denotes the current mode at time τ and f_a is the mode to be inserted, then it is an elementary exercise to show that the variation in $x(\tau)$ at time τ due to this insertion is give by

$$\frac{dx_\tau}{d\tau} = f_a(x(\tau)) - f_{\sigma(\tau)}(x(\tau)) \tag{33}$$

The cost gradient due to the insertion of a mode f_a at point τ can be written from chain rule as

$$\frac{dJ}{d\tau}(f_a) = \frac{dJ}{dx_\tau}\,\frac{dx_\tau}{d\tau} \tag{34}$$

If we plug the expression from (32) and (33) into the above equation, we get

$$\frac{dJ}{d\tau}(f_a) = -p(\tau)^\top \left(f_a(x(\tau)) - f_{\sigma(\tau)}(x(\tau))\right) \tag{35}$$

which is the same as equation (35) except for the negative sign to signify the descent direction and thus we have arrived at the same equation derived in [25, 26]. Thus the vector difference $f_\Delta(x(\tau)) = \left(f_a(x(\tau)) - f_{\sigma(\tau)}(x(\tau))\right)$ in the equation (35) encodes information about the change in initial condition and we can look at the insertion gradient as the inner product

$$\frac{dJ}{d\tau}(f_a) = \left\langle p(\tau)\,,\, f_\Delta(x(\tau))\right\rangle \tag{36}$$

Whenever, the vectors $f_\Delta(x(\tau))$ points in a direction belong to the opposite half of the normal drawn to the vector $p(\tau)$ at the point $x(\tau)$, the inner product and hence the insertion gradient is negative indicating a decrease in cost due to insertion of mode f_a and vice versa.

Fig. 17 - 19 demonstrate the above key observations via the example of *Two Switched System* introduced in the previous chapter. Fig. 17 shows the values of

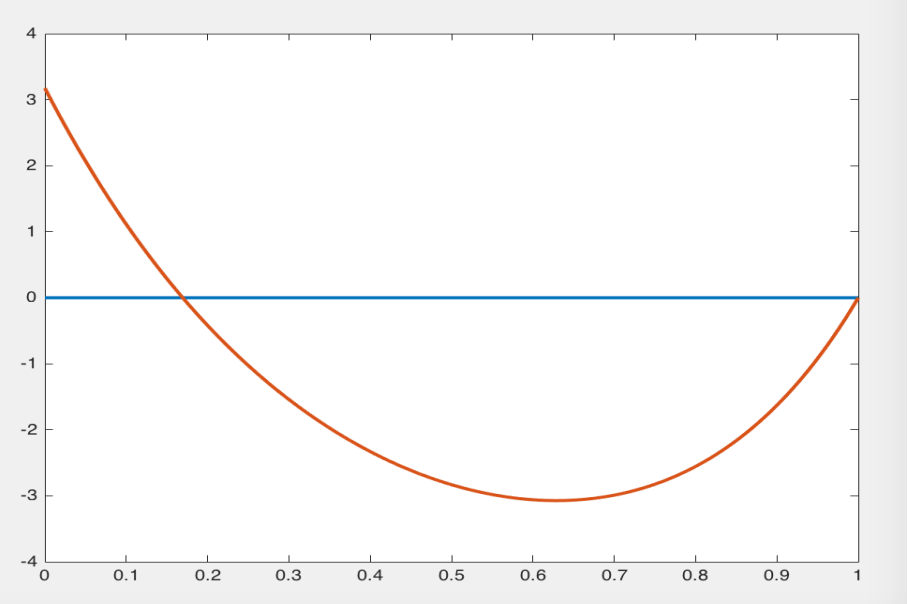


Figure 17: Mode Insertion Gradient for the two mode switched system for the case when mode 1 is the active mode. The points where the insertion gradient is negative indicates the region where the insertion of mode 2 for small duration can result in a decrease in cost.

insertion gradients computed from equation (30) for the case when only mode 1 is active. The regions, where the insertion gradient is negative are the set of points where the insertion due to mode 2 an any of these points will result in a decrease in cost. Fig. 18 shows the geometric picture where we have plotted the state trajectory $x(t)$ in the plane and the vectors $p(t)$ and $f_\Delta(x(t)) = f_2(x(t)) - f_1(x(t))$ at the respective points $x(t)$ corresponding to different values of t. The state trajectory and costate trajectory correspond to the active mode 1. Whenever the relative angles between the two vectors at the respective points along the state trajectory is between $(90^0, 270^0)$, as shown in the polar plot Fig. 19, the corresponding insertion gradient in Fig. 17 is negative and vice versa.

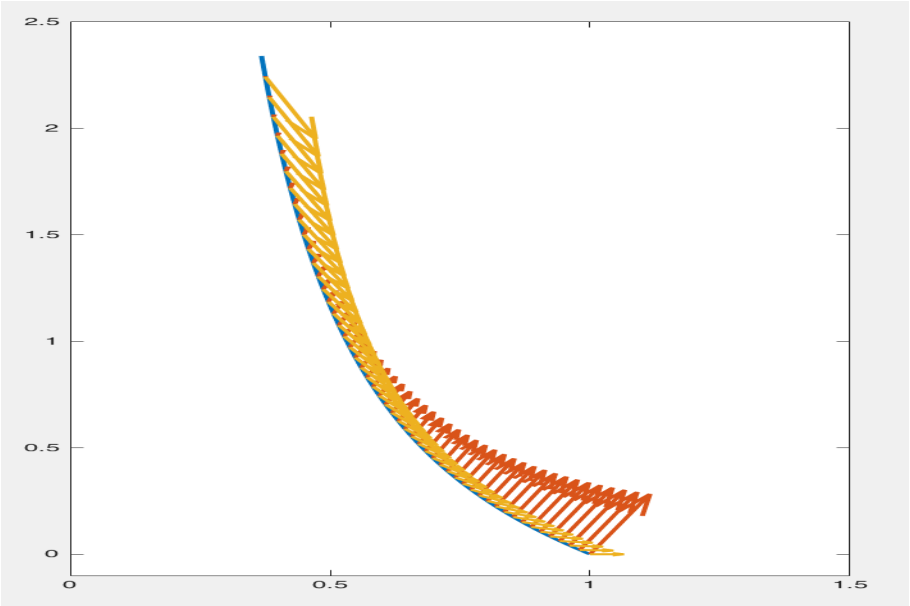


Figure 18: Figure shows the state trajectory x in the plane $x_1 - x_2$ and the vectors p and f_Δ at every point along the state trajectory. At points, where the relative angle between the vectors is between $(90^0, 279^0)$ are the points where the insertion gradient is negative in Fig. 17.

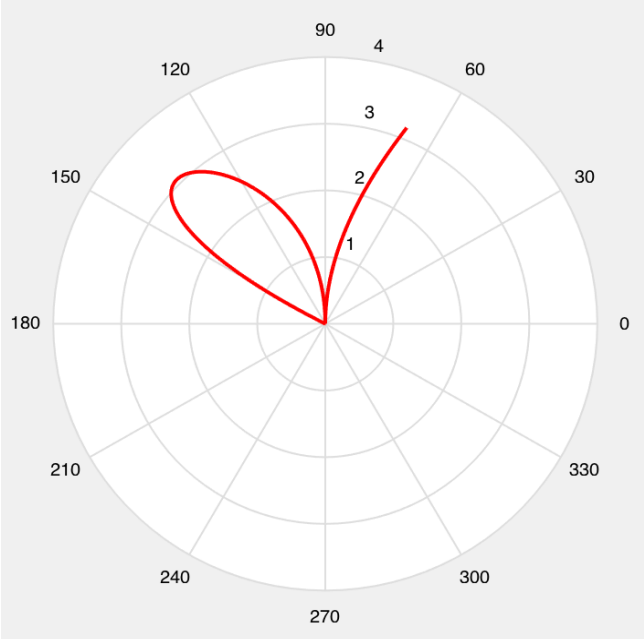


Figure 19: Figure shows the relative angles between the costate vectors p and f_Δ at each point along the state trajectory in Fig. 18. The radial length shows the magnitude of the inner produce and hence the insertion gradient, and the angle signify the angle difference between the two vectors.

4.2 Mode Insertion Gradient: Connections with Maximum Principle

While the above description gives us a look into geometric description, is there any relationship between the Maximum Principle and the mode insertion gradient? This is the question what we consider next. Let $H_{v(t)}$ denotes the hamiltonian associate with the current mode schedule and $H_a(t)$ be the hamiltonian associated with the mode that we are checking for insertion. Then

$$H_{v(t)} = p(t)^\top f_{v(t)}(x(t)) + L(x(t)) \tag{37}$$

$$H_{a(t)} = p(t)^\top f_{a(t)}(x(t)) + L(x(t)) \tag{38}$$

If we take the difference of the above two Hamiltonians, we get

$$H_\Delta^{a(t)} = p(t)^\top \left(f_{a(t)}(x(t)) - f_{v(t)}(x(t)) \right) \tag{39}$$

which is the same as equation (30). Now let

$$a^*(t) = \min_{j \in \mathcal{Q}} \quad \{H_\Delta^{j(t)}\} \tag{40}$$

and

$$t^* = \min_{t \in [t_0, t_f]} \quad \{H_\Delta^{a^*(t)}\} \tag{41}$$

Then

$$\Theta = H_\Delta^{a^*(t^*)} \tag{42}$$

is the optimality function that is being utilized in [10, 25] and others to characterize the optimality condition. The argument $(t^*, a^*(t^*))$ is used for updating the mode sequence at each iteration. This function takes on only negative values and this being 0 is the optimality condition, which coincides with the maximum principle, since we have then the optimality condition

$$H(x, p, v^*(t), t) \leq H(x, p, v(t), t) \tag{43}$$

for all the switching signals $v(t)$ in the optimization space $\mathcal{V}$ along the optimal trajectories. Notice here that $v(t)$ is our control variable that selects one of the $\mathcal{Q}$ modes, which in the absence of continuous external input is the function of state variable. In other words, we could just push the subscript inside the vector field, denote it by u and we get the usual maximum principle. The above representation is very useful since it immediately allows us to extend this optimality function to the case when both the continuous and switching controls are present.

For the case of controlled switched systems, where the subsystem dynamics depend on the continuous external input u as well as the switching signal $v(t)$, the control vector becomes $w(t) = (v(t),\, u(t))$ and the overall system dynamics can be written as

$$\dot{x}(t) = f_{v(t)}\big(x(t),\, u(t)\big) = f\big(x(t),\, w(t)\big), \qquad x(0) = x_0 \tag{44}$$

and the common optimal control problem is to minimize the cost functional

$$J = \int_{t_0}^{t_f} L(x(t),\, w(t))\, dt + \phi(x(t_f)) \tag{45}$$

subject to the dynamical and initial condition constraints (44) and where u belongs to some bounded set. If we denote the Hamiltonians associated with the current control by H_w and the one associated with the insertion of mode a and continuous control z by H_ζ where $\zeta(t) = (a(t), z(t))$, then

$$H_{w(t)} = p(t)^\top f_{v(t)}(x(t),\, u(t)) + L_{v(t)}(x(t),\, u(t)) \tag{46}$$

$$H_{\zeta(t)} = p(t)^\top f_{a(t)}(x(t),\, z(t)) + L_{a(t)}(x(t), z(t)) \tag{47}$$

If we take the difference of the above two Hamiltonians, we again get mode insertion gradient, which in this case is given by

$$H_\Delta^{\zeta(t)} = p(t)^\top \Big(f_{a(t)}(x(t),\, z(t)) - f_{v(t)}(x(t),\, u(t)) \Big) + \Big(L_{a(t)}(x(t), z(t)) - L_{v(t)}(x(t),\, u(t)) \Big) \tag{48}$$

The optimality function to characterize the optimality condition is then given by

$$\Theta = H_{\Delta}^{\zeta^*(t^*)} \tag{49}$$

where

$$\zeta^*(t) = \min_{\substack{j \in \mathcal{Q} \\ z \in \mathcal{B}}} \quad \left\{ H_{\Delta}^{\zeta} \right\} \tag{50}$$

and

$$t^* = \min_{t \in [t_0,\, t_f]} \quad \{ H_{\Delta}^{\zeta^*(t)} \} \tag{51}$$

gives the mode to be inserted and the point of insertion. References [36, 37] gives rigorous proofs for the derivation of the optimality function and a numerical scheme for computing the optimal controls with constraints on the state trajectory. We see here, however, how using our Hamiltonian formalism for the mode insertion gradient results in the easy extension of the results for autonomous systems to controlled switched optimal control problems.

For the case when the continuous external input is absent, notice that the minimizer $f_{a^*(t)}$, gives the point wise minimizer of the Hamiltonian at any given iteration, while the optimality function (42) just inserts the mode f_a^* at a single point t^* and the new mode sequence is subsequently optimized by performing switch time optimization which makes the algorithm quite inefficient. In [76–78], the authors devised a powerful method that can utilize the information on $f_{a^*}(t)$ to update the mode sequence over a measurable set of points, thereby eliminating the need for switch time optimization. The set of points over which $f_{\sigma(t)}$ is replaced by $f_{a^*(t)}$ is chosen such that, using our notation of Hamiltonians, has the property that

$$S = \{ t \in [t_0,\, t_f] \quad | \quad H_{\Delta}^{a^*(t)} \leq -\eta \}, \tag{52}$$

with the modes being inserted given by $f_{a*(S)}$. The algorithm convergence was then proved using the property of uniform sufficient descent [76–78], using the optimality conditions defined in terms of the optimality function defined via equation (42).

With continuous controls, one can update the control by using

$$\begin{aligned} u_{next}(t) &= u(t) + \lambda(u^*(t) - u(t)) \\ &= (1-\lambda)u + \lambda u^* \end{aligned} \tag{53}$$

where $\lambda \in [0, \lambda_{max}]$ is a constant chosen according to some rule, like Armijo Rule [9] and u^* being the minimizer of Hamiltonian. With the discrete set of controls such as our switching control $v(t)$, we can write by extension

$$\begin{aligned} f_{v_{next}(t)} &= f_{v(t)} + \lambda(t)(f_{v^*(t)} - f_{v(t)}) \\ &= (1-\lambda(t))f_{v(t)} + \lambda(t)f_{v^*(t)} \end{aligned} \tag{54}$$

with $\lambda \in \{0, 1\}$, effectively implying, either insert the mode when $\lambda = 1$, so that the above equation becomes $f_{v_{next}(t)} = f_{v^*(t)}$, or keep the existing mode when $\lambda = 0$, in which case the above equation becomes $f_{v_{next}(t)} = f_{v(t)}$, and this choice of λ needs to be made for each t. This problem stems from the discrete nature of the problem, where the control can take only fixed values. In the particular case where only one mode is inserted at a single point such as in [10, 25] etc, the condition on λ is that $\lambda(t) = 0$ for all $t \in \{[t_0,\, t_f] - t^*\}$ and $\lambda(t^*) = 1$. The discussion holds however true when the continuous control input u is also present

Looking at equations (53) and (54), one can see that the next control is a convex combination of the current control and the minimizer of the Hamiltonian, if λ is allowed to take values in the interval [0, 1]. Methods based on this convex relaxation has been used to take advantage of the entire f_{v^*}, which is the point-wise minimizer of the Hamiltonian at the current iteration, and single step size λ in the descent direction computed using some step size method as Armijo rule have been used in [12, 38] for approximating solutions to the switched optimal control for autonomous as well as non-autonomous switched affine systems with convex costs.

All of the above is possible since we are inserting the mode at a point in the limit, and the state and costate trajectories being computed from the integration of

differential equations remain unaltered due to this needle insertion. Optimization algorithms based on mode insertion gradient take advantage of this underlying fact, since the state and costate trajectories do not need repetitive computation for finding the optimal mode insertion in order to update the mode sequence. The fundamental problem with optimizing mode sequence for the dwell time problem arises fundamentally because of this problem since the mode has to be inserted for some duration rather than a point [5] and is the reason why algorithms that employ directly the precomputed mode insertion gradient as in [77] are bound to fail in general for arbitrarily small dwell times.

In chapter 3, we presented how to deal with the switch time optimization part of the dwell time problem effectively and presented a conceptual algorithm for mode sequence optimization part. In the next section, we present an effective method based on mode insertion gradients for solving the second part of the puzzle and thus we will have an implementable algorithm for the dwell time problem by the end of the section. We will consider several examples for bot linear and nonlinear systems to demonstrate the algorithm.

4.3 Mode Sequence Optimization

In this section, we explain how to utilize the information of mode insertion gradient for updating the mode sequence, while satisfying the dwell time constraints i.e. the insertion should be feasible. One can insert a new mode at the point in feasible insertion region corresponding to the minimum insertion gradient as in [10, 25] by introducing two new switch times ϵ apart around it and then perform switch time optimization over the two new switch times introduced ϵ apart, while holding the rest of switching times fixed. If the two switch times are at least δ apart at the end of this optimization, we have the updated mode sequence which can then be switch time optimized subject to dwell time constraints. However there are two

problems associated with this procedure. As pointed out in [5], it can happen that after performing optimization over the two new switching times, are not δ apart, and one would tend to reject that mode insertion. However, if one inserts the new mode for the duration of dwell time in the neighborhood of the point corresponding to minimum insertion gradient, there can still be a decrease in cost, and it is not evident whether to accept or reject the new mode. Secondly, this approach for new mode insertion is inefficient in the sense, that the state and costate trajectories need to be computed repeatedly during the optimization process. Moreover integrating the insertion gradient as in [77] over any significant dwell time to compute the change in cost does not help owing to the fact that the state and costate trajectories and hence the insertion gradient changes as a function over the duration of the dwell time. This can be seen in Fig. 20 for the *'Two Switched System'* introduced in chapter 3, where the insertion gradient is plotted corresponding to different dwell times. The 4 different curves correspond to inserting mode 2 for duration of 0s, 0.001s, 0.01s and 0.1s around the point corresponding to minimum insertion gradient $t = 1.2066$s. As can be seen from the figure, as the duration of insertion of the new mode increases, the insertion gradient for mode 2 departs significantly from the original insertion gradient in the regions where mode 1 is active. In some cases, insertions of even small durations can produce significant changes in the mode insertion gradients. For example adding mode 1 to the switch time optimized mode insertion gradients in Fig. 21 at $t = 0.9$s for duration of dwell time $\delta = 0.01$s results in the new insertion gradients shown in Fig. 22 that are drastically different from the original ones. In particular note the lift in the insertion gradient d_1 around the point of insertion, becoming positive, indicating the new insertion might not result in a decrease in cost. Having identified the issues, we next proceed towards presenting techniques to obviate those problems. We start off with the following lemma that captures the information in this paragraph regarding the variation in insertion gradient as a function of duration of dwell time.

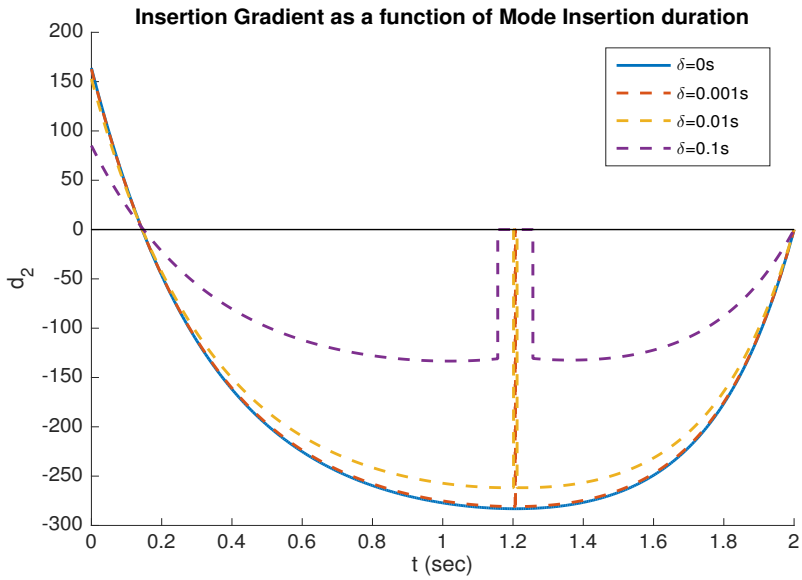


Figure 20: Figure depicts the variation in the insertion gradient due to inserting mode 2 for the duration of dwell time around feasible insertion point corresponding to $t = 1.2$s. The insertion gradient due to mode 1 is not shown to avoid clutter. The insertion gradient due to mode 2 is zero in the region where mode 2 is inserted for the duration of dwell time.

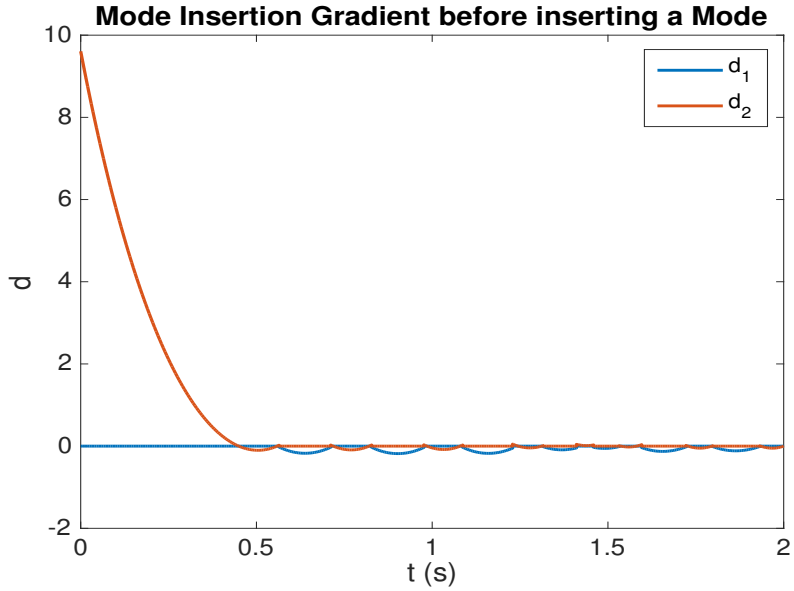


Figure 21: Mode insertion gradients of the switch time optimized mode sequence. As we get close to the optimality, the number of switches increases and the insertion gradients in magnitude get closer to zero.

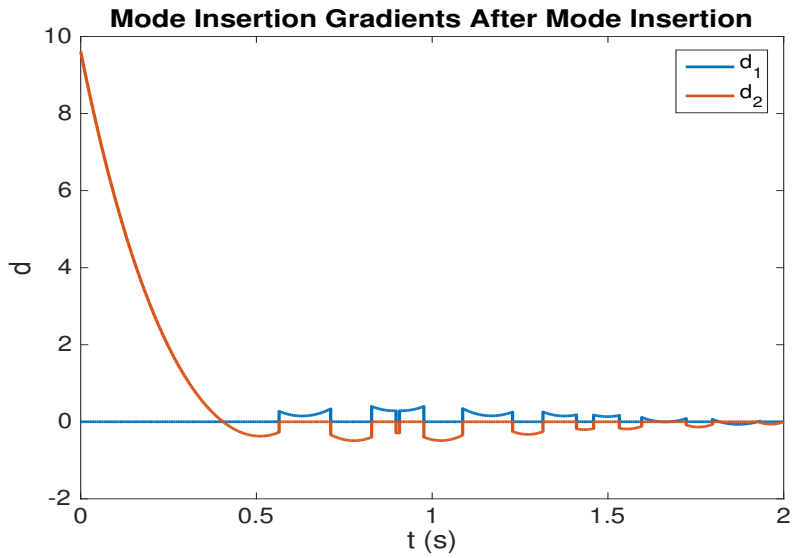


Figure 22: Mode insertion gradients after addition of mode 1 to the switch time optimized mode sequence at the point corresponding to minimum over all insertion gradients, $t = 0.9$s for duration of dwell time $\delta = 0.01$s. Note the large variation in the curves of insertion gradients due to this small insertion. In particular, the insertion gradient due to mode 1 is positive, indicating, that insertion even for such small duration might not be acceptable.

Lemma 2 *The mode insertion gradient d is a continuous function of the switching times τ.*

Proof 3 *The mode insertion gradient due to mode a at any time t is given by $d_a = p(t)\left(f_a(x(t)) - f_{\sigma(t)}(x(t))\right)$. By assumption, $f_\sigma(x)$ is continuously differentiable with respect to x and since x and p are continuous function of τ, we have a product of two continuous functions which is continuous. If the two switch time vectors are close in l_1 norm, the corresponding insertion gradients are close by in the L_∞ norm.*

The importance of this lemma is that if we start at a negative minima and increase the duration of insertion of the mode from 0 to the δ and the insertion gradients are still negative at the two switching times, the insertion is valid since pushing the switching times away from each other will result in a further decrease in cost which brings us to our new lemma.

Suppose a mode b is inserted to an existing mode sequence for the duration of dwell time δ in the feasible insertion region and centered around the point corresponding to minimum insertion gradient. Suppose D_b^- and D_b^+ are the values of the new insertion gradient defined to the left and right of the two new switch times introduced due to insertion of mode b, respectively. Then

Lemma 3 *If $D_b^- < 0$ and $D_b^+ < 0$, inserting mode b to the existing sequence results in lowering the cost*

Proof 4 *The insertion gradient d_b is the change in cost due to inserting the mode at a point. The fact that it is negative to the left and right of mode b means, the two switch times can be shifted further away from each other resulting in further lowering of cost. Hence the result*

The above lemma is a key to the fast implementation of the mode updating stage of our algorithm, since if the condition holds, we can immediately insert the new mode and go to switch time optimization stage. It gives us the power to immediately check for the validity of a new insertion for updating mode sequence, without requiring to perform any switch time optimization. One needs to compute the state and costate trajectories to get the new insertion gradient corresponding to the new mode sequence only. The insertion gradients D_b^- and D_b^+ need not necessarily be the same and thus the insertion location need not be the one that gives the maximum decrease in cost. Instead, we let the switch time optimization of stage A to take care of the optimal placement of the mode by optimizing the overall switching times subject to dwell time constraints.

To illustrate the implication of the above lemma, we again consider the *'Two Switched System* example considered previously where mode 1 is active and we want to check for the insertion of mode 2 using the criterion mentioned in the above lemma. Fig. 23 shows the insertion gradients for the two modes when only mode 1 is active

while Fig. 24 depicts the insertion gradients corresponding to the insertion of mode 2 centered at $t = 1.2$s, the point where the insertion gradient in Fig. 23 is minimum. As can be seen in Fig. 24, the insertion gradients D_b^- and D_b^+ are both negative, indicating the feasibility of this insertion for updating the mode sequence. After performing the switch time optimization over the new sequence, the insertion gradients for the two modes are shown in Fig. 25, indicating the two switch times have indeed moved away. At the optimal solution, the insertion gradients at the switch times have become zero, which is the case when no dwell time constraints are active.

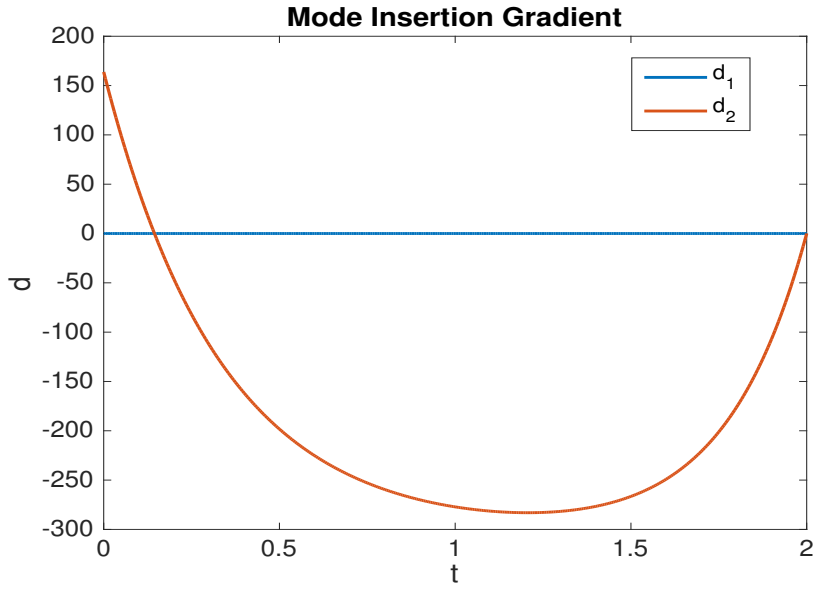


Figure 23: Mode insertion gradients after the sequence before inserting a new mode. Insertion gradient due to mode 1 is zero since it is active over the entire horizon.

Next we consider the case when we don't have the situation mentioned above i.e. either D_b^- or D_b^+ or may be both are non-negative. Introducing the new mode placed symmetrically about the point corresponding to the minimum over the insertion gradients is not necessarily optimal. For example, as can be seen in the Fig. 26, the mode placement that gives a maximum decrease in cost when inserted for the duration δ is not symmetrical about the insertion point, corresponding to the minimum over insertion gradients. In such a case, we start with the new switch times

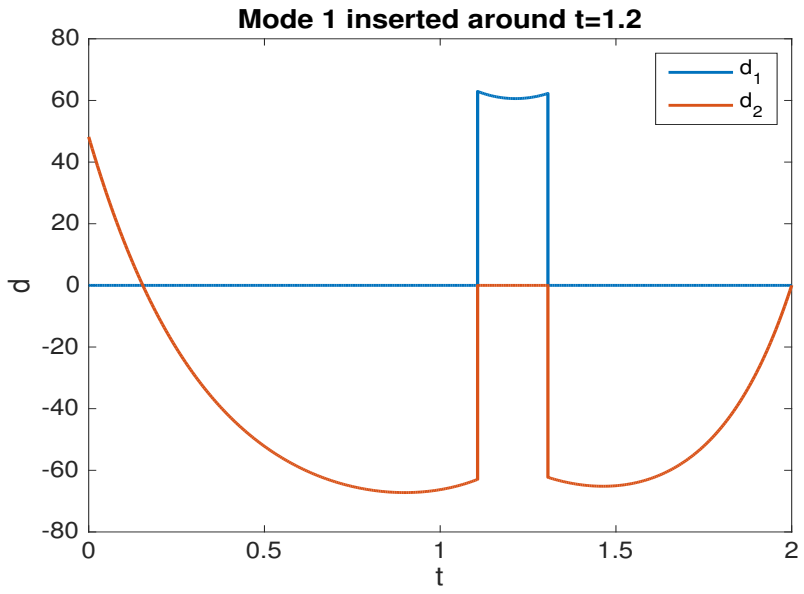


Figure 24: Mode insertion gradients after the sequence is modified by insertion of new mode for duration of minimum dwell time around the time point corresponding to minimum insertion gradient.

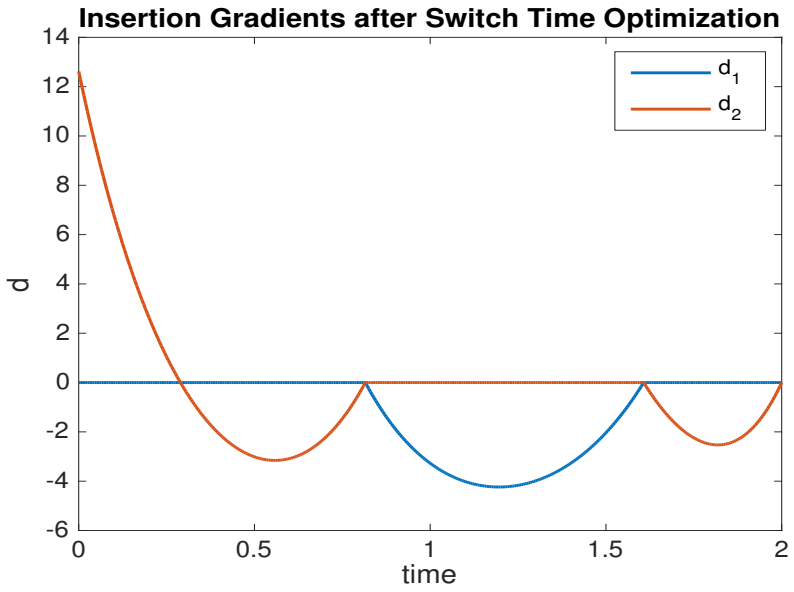


Figure 25: Mode insertion gradient after performing switch time optimization. The insertion gradients are zero at the optimal transition points between the two modes.

as mentioned above and compute the gradients of the cost w.r.t the new insertion point. Suppose the mode under consideration is f_b which is inserted to the existing sequence for duration of dwell time between t_1 and $t_2 = t_1 + \delta$, then keeping the other switching times fixed, it is an elementary exercise in calculus of variation to show that the gradient of the cost w.r.t. t_1 subject to the fixed dwell time constraint is given by

$$\frac{dJ}{dt_1} = \frac{1}{2}\left[p(t_1)\left(f_{\sigma(t_1^-)}(x(t_1)) - f_b(x(t_1))\right) + p(t_2)\left(f_b(x(t_2) - f_{\sigma(t_2^+)}(x(t_2))\right)\right] \quad (55)$$

where $x(t)$ and $p(t)$ are the state and costate equations associated with the modified mode sequence and the new switch time vector incorporating the two newly introduced switching times. The above gradient expression can be viewed as the average of the individual cost gradients w.r.t switching times t_1 and t_2 respectively and at the point of optimality, where this gradient becomes zero, we have the condition that

$$d_b(t_1) = d_b(t_2) \quad (56)$$

where d_b is the insertion gradient due to mode b. This can be seen in Fig. 27 where the mode insertion gradient due to mode 1 becomes equal at the two switching times, corresponding to the optimal positioning of switching function subject to fixed dwell time of $\delta = 0.2$ sec and using Armijo step size rule in the descent direction given by (55). Note that the optimal position of the mode is not symmetrical about the line corresponding to minimum insertion gradient in Fig. 26

So, we descend in the direction of gradient using Armijo step size rule for updating the two switch times while holding the others fixed till either of the following happens sequentially

- D_b^- and D_b^+ both become negative, in which case we have found the mode to be inserted by Lemma 2.
- We reach the optimal solution and check the cost differential (25), which if

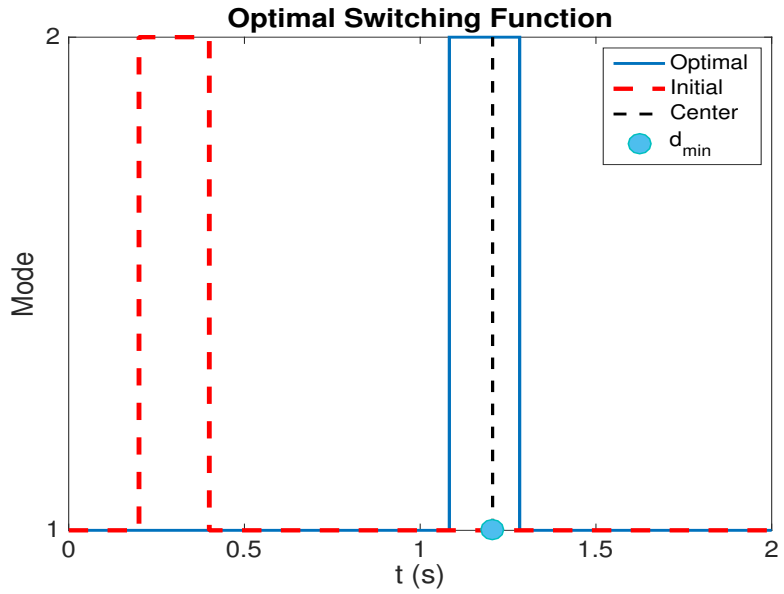


Figure 26: Switching function before (red) and after (blue) optimization. The optimal mode inserted for duration of dwell time $\delta = 0.2$s is not symmetrical about the point corresponding to minimum insertion gradient.

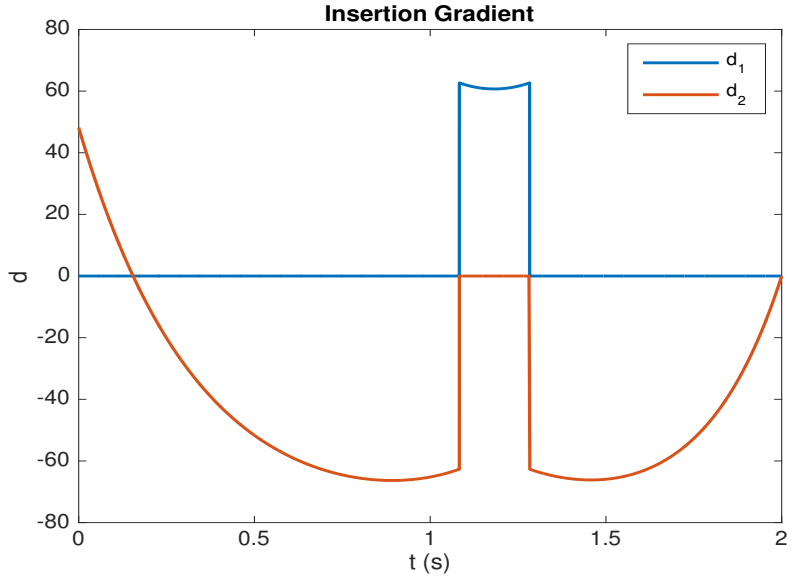


Figure 27: Mode insertion gradient after performing switch time optimization subject to fixed dwell time for mode 2. The insertion gradients are equal at the optimal transition points.

negative, allows us to insert the mode, even though the conditions of Lemma 2 are not applied.

- If the above two situations don't happen, we disregard the mode and repeat the process with other local minima's (mode to be inserted, time of insertion) of the mode insertion gradient.

- If no new mode can be inserted, we have reached the optimal solution, since the optimality function (15) becomes zero. If a mode insertion is found, the mode sequence is switch-time optimized subject to dwell time constraints as described in chapter 3.

The fact that we just need the first instance or inkling of the mode to result in a decrease in cost to update the mode sequence really speeds up the optimization process for mode insertion, since we don't necessarily need to reach the optimal solution. Moreover, the insertion gradient values need computation only at the new switching times rather than the entire interval $[0,\, t_f]$. We also get rid of the problem, in regards to making the decision about accepting or rejecting a mode, that was mentioned earlier, by the above approach. Next we formalize the mode sequence update procedure mentioned in the preceding lines.

For a switch time optimized mode sequence σ satisfying dwell time constraints, let U denote the feasible insertion region as shown in Fig. 12 in chapter 3. Let the insertion gradient due to mode j be denoted by d_j which is given by

$$d_j(t) = p(t)\left(f_j(x(t)) - f_{\sigma(t)}(x(t))\right) \tag{57}$$

where $x(t)$ and $p(t)$ are the state and costates associated with the mode schedule (σ, τ). Let

$$[t_{ins},\, f_{ins}] = \arg\min\Big\{ d_j(t) \quad | \quad t \in U, \quad j \in \mathcal{Q} \Big\}, \tag{58}$$

where

$$U = \bigcup_{k=1,j=1}^{N,Q} (I_{k,j} \cup P_{k,j}) \tag{59}$$

In this representation, the intervals are duplicated, but it helps us in the search for updating the mode sequence, since we eliminate the set $(I_{j,k} \cup P_{k,j})$ to which $f_j = f_{ins}$ belongs whenever the insertion is unsuccessful, in the sense that it does not reduce the cost. The algorithm is terminated when this set becomes an empty set, indicating that the optimality function θ as defined in (15) has become zero, since no reduction in cost can be obtained due any mode insertion. So if we let $\mu(U)$ then denote the Lebesgue measure of set U, then we have the equivalent optimality coditions

$$\mu(U) = 0 \iff \theta = 0 \tag{60}$$

We assume that each mode insertion gradient takes one minima in any interval and we start with the smallest. However, it can be easily generalized to the case of multiple minimas. If it fails, we remove that interval, choose the next small one over the remaining intervals and repeat the process, till either a mode insertion has been found, or terminate the algorithm when no mode can result in a lower cost.

There is one more issue however that needs to be addressed before we formally state the algorithm for mode sequence optimization and that is how to place the mode for duration of dwell time around the point of insertion t_{ins} corresponding to the minimum over the insertion gradients, since the insertion needs to be feasible. In other words, the mode inserted should not come close to the existing modes by less than dwell time and it can happen that the minima happens to be within, or very close to non-feasible regions, as shown in Fig. 12. We would like to place the new mode symmetrically centered at t_{ins} when dwell times are respected. However, we have to make a decision, when this is not the case. We choose the two new switching times according to the following criterion.

$$
\begin{aligned}
&t_1 = t_{ins} - \frac{\delta}{2} \quad \text{and} \quad && t_2 = t_{ins} + \frac{\delta}{2} \quad && \text{if} \quad && t_{ins} \in \left[\tau_k + \frac{3}{2}\delta,\, \tau_{k+1} - \frac{3}{2}\delta\right], \\
&t_1 = \tau_k \quad \text{and} \quad && t_2 = \tau_k + \delta \quad && \text{if} \quad && t_{ins} \in [\tau_k + \delta,\, \tau_k + \frac{3}{2}\delta), \\
&t_1 = \tau_{k+1} - \delta \quad \text{and} \quad && t_2 = \tau_{k+1} \quad && \text{if} \quad && t_{ins} \in (\tau_{k+1} - \frac{3}{2}\delta,\, \tau_{k+1} - \delta], \\
&t_1 = t_{ins} \quad \text{and} \quad && t_2 = t_{ins} + \delta \quad && \text{if} \quad && t_{ins} = \tau_k
\end{aligned} \tag{61}
$$

where the last three scenarios correspond to the cases when the time t_{ins} corresponding to the minima of insertion gradients belong to the infeasible regions. Next, we describe the Mode Insertion Algorithm.

Algorithm II - Mode Sequence Update

Assuming a switch time optimized mode sequence σ satisfying dwell time constraints, compute the mode insertion gradients due to each mode via (57) and do the following:

1. Compute the next t_{ins} and f_{ins} via (58) and find the insertion points t_1 and t_2 via (61).

2. Update τ with the two new switching times t_1 and t_2 and compute $D_{f_{ins}}(t_1^-)$ and $D_{f_{ins}}(t_2^+)$. If both become negative, stop and insert the mode.

3. Update t_1 and t_2 only in τ by descending in the gradient direction (55) according to Armijo rule and compute $D_{f_{ins}}(t_1^-)$ and $D_{f_{ins}}(t_2^+)$. If at any iiteration, both become negative, stop and insert the mode. Otherwise at the optimal point, compute J_Δ via (25). If $J_\Delta < 0$, insert the mode and stop. Otherwise, make the interval $I_{k,j} \cup P_{k,j}$ to which the minima t, f_{ins} belong an empty set and update the search region U in (59) for computing the next minimum.

4. If U is an empty, then set $\theta = 0$ to terminate the algorithm. Otherwise go to step 1.

With both the switch time optimization and mode sequence optimization implementations available, we can now state our dwell time algorithm.

Dwell Time Algorithm - II

Given a mode sequence σ having n modes and vector τ satisfying the dwell time constraints and time optimized, identify the feasible region U as defined in (59) and do the following

1. If the Lebesgue measure of the set U i.e. $\mu(U) = 0$ then **Stop**.

2. Insert mode f^* in the interval $[t, t+\delta]$ where the pair (t^*, f^*) is obtained from the mode sequence optimization i.e. *Algorithm II* where the update of U also occurs. We get new mode sequence σ and switching time vector τ.

3. For the new mode sequence σ, solve the switch time optimization problem using *gradient projection algorithm* to get optimal τ and go to step 1.

4.4 Convergence

In chapter 3, we introduced the notion of sufficient descent in the context of proving algorithm convergence for the conceptual dwell time problem. Since $J(\xi) \geq 0$ for all $\xi \in \mathcal{X}$, the algorithm having the property of sufficient descent converges i.e for every infinite sequence $\{\xi_j\}$ computed by algorithm, $\lim_{j\to\infty} \theta(\xi_j) = 0$. Otherwise $\lim_{j\to\infty} J(\xi_j) = -\infty$ which contradicts the fact that J is bounded from below. The convergence is obvious if $\{\xi_j\}$ is a finite sequence.

Lemma 4 *The dwell time algorithm-II has the property of sufficient descent.*

Proof 5 *Consider our optimality functions (15) and (60). Since* $\theta_{\mathcal{X}}(\xi_j) = J(\xi_{j+1}) - J(\xi_j)$ *and* $\theta_{\mathcal{X}}(\xi_j) < -\delta'$ *with* $\delta' > 0$ *at all non-stationary points, we have* $J(\xi_{j+1}) - J(\xi_j) < -\eta$ *with* $0 < \eta = \delta'$. *Otherwise* $\mu(U) = 0$ *and by equivalence (60),* $\theta = 0$, *indicating we are at the optimal solution, a contradiction. Hence proved.*

Again, since our optimization space is finite dimensional, we do not run into problems associated with infinite dimensional parameter spaces [10] to prove the sufficient descent property of our algorithm. We now state the main theorem about algorithm's convergence.

Theorem 2 *The dwell time algorithm converges to stationary points of the optimality function $\theta_{\mathcal{X}}$.*

Proof 6 *The proof follows immediately from the property of sufficient descent proved for our algorithm and the discussion following the definition of sufficient descent regarding algorithm convergence.*

.

4.5 Simulation Results

In this section, we present several examples for both linear and nonlinear switched systems and costs under different dwell time constraints. We begin with the linear case.

4.5.1 Linear Case

We consider the same *'Two Switched System'* considered in the previous chapter, whose dynamics are governed by the matrices

$$A_1 = \begin{pmatrix} -1 & 0 \\ 1 & 2 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & 1 \\ 1 & -2 \end{pmatrix}$$

corresponding to modes f_1 and f_2 respectively and a quadratic cost defined on the state trajectory as

$$J = \frac{1}{2}\int_0^{t_f} ||x(t)||^2 dt.$$

The switched system is initialized to $x_0 = (1, 0)'$ and optimized over time interval $[0, 2]$. We assume both the modes have the same minimum dwell time. The Armijo

parameters are set to $\alpha = 0.5$ and $\beta = 0.5$. The time step for simulation was set to $dt = 1e^{-3}$ s. For intervals in the set $U \subset [0, T]$ for feasible mode insertion, the time step used was $ds = 1e^{-3}$ s. The mode sequence is initialized to $\sigma_0 = \{f_1\}$ for which the cost is $J = 40.75$. The algorithm is terminated when $|\theta_{\mathcal{X}}(\cdot)|$ falls below $\epsilon = 0.1e^{-3}$. The simulation is performed for dwell time constraints of 0.01 s and 0.1 s and results are shown in Fig. 28 and Fig. 29 respectively. To emphasize the variation of cost with each iteration, the initial cost $J = 40.75$ which is relatively large is not shown for the cost trajectory and the graph starts with iteration 1. For the case where the dwell time is $0.01s$, it takes about 9 iterations before the optimality conditions are met and the final cost is $J = 2.39$. For the second case, it takes about 3 iterations before the optimality conditions are satisfied. The final cost $J = 2.60$ is relatively higher than with smaller dwell time constraints as expected. Also due to the presence of larger dwell time constraints in the second case, some of the modes that appear in the optimal mode schedule of Fig. 28 do not appear in optimal mode schedule of Fig. 29.

4.5.2 Nonlinear Case - Double Tank Problem

Next we consider the case of the double tank problem considered in [76] for the application of our algorithm. The problem is depicted in Fig. 30 where the objective is to regulate the level of water in the second tank by operating the switch that controls the flow of water to the first tank. Here the dynamics of the nonlinear system are given by the Torricelli's law

$$\begin{aligned} \dot{x}_1 &= k - \sqrt{x_1} \\ \dot{x}_2 &= \sqrt{x_1} - \sqrt{x_2} \end{aligned} \tag{62}$$

where k is an integer either taking the value 1 or 2, so that the system can switch between the two dynamics corresponding to each value of k. The cost is chosen so as regulate the level of water in tank 2 to level 3 by switching between the two modes.

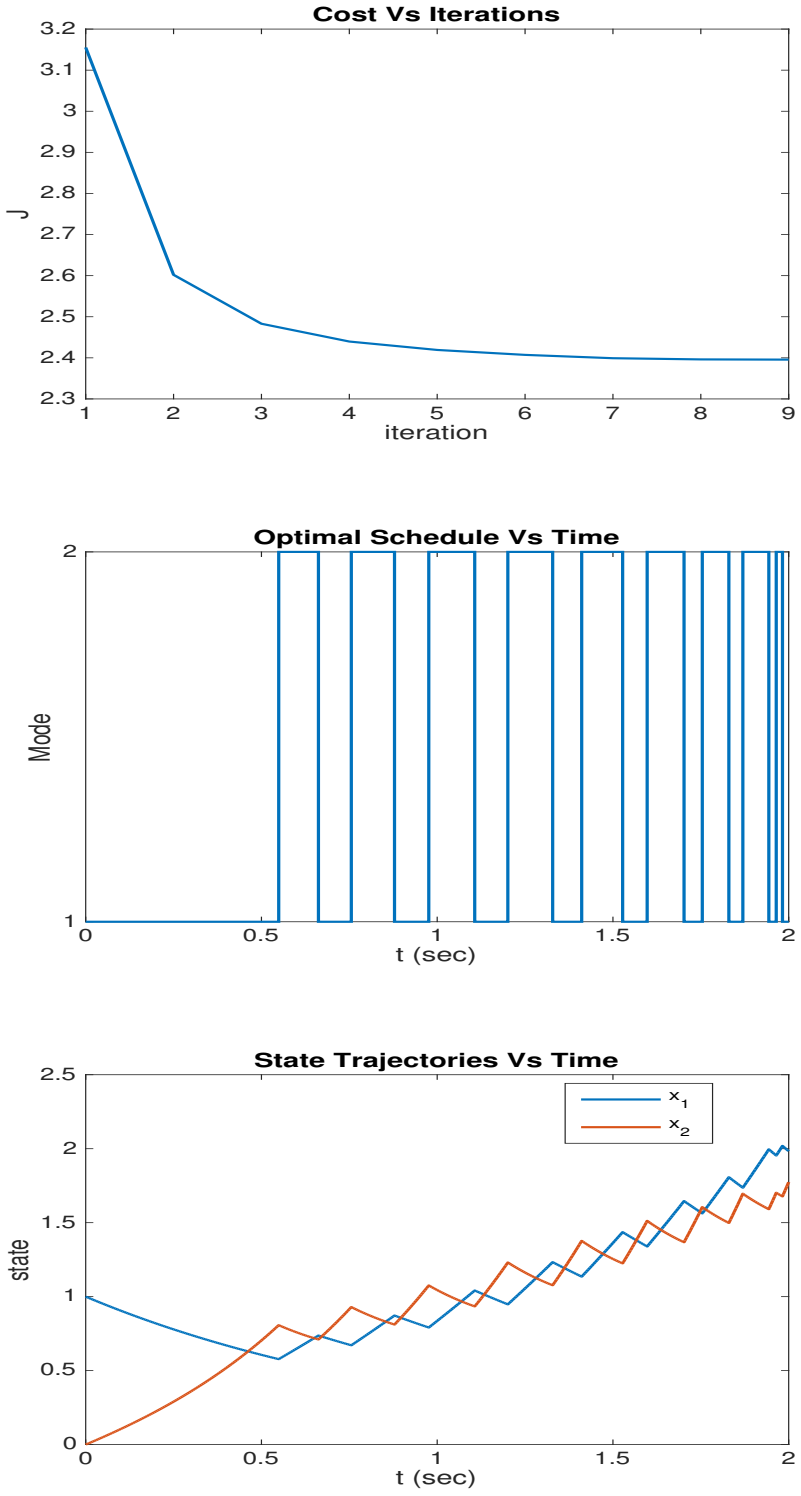


Figure 28: Plot of cost versus iterations, optimal mode schedule and state trajectories for $\delta = 10ms$. The algorithm stops after iteration 9 when no new mode can be inserted and the optimality conditions are satisfied. The abrupt changes in state trajectory correspond to the switching between modes.

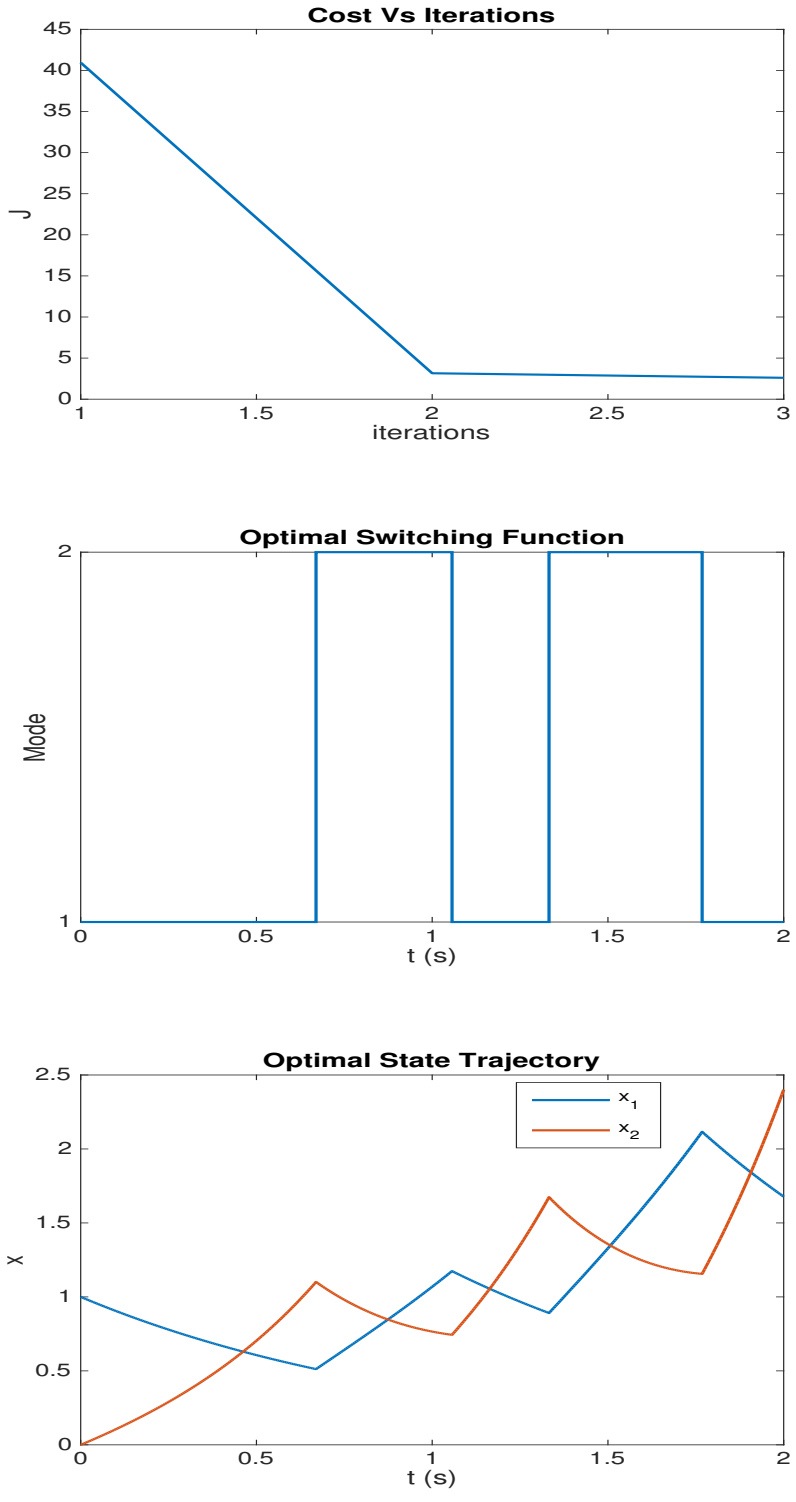


Figure 29: Plot of cost versus iterations, optimal mode schedule and state trajectories for $\delta = 100ms$. Because of the relatively large dwell time, the number of switches has reduced and the optimal cost is also higher than with small dwell times, as expected.

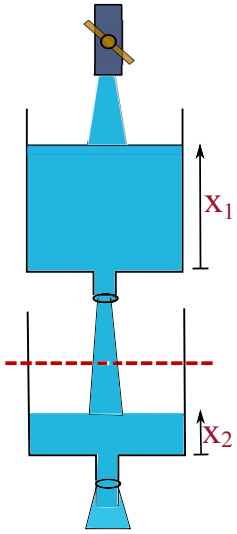


Figure 30: Double Tank Problem. The level of water in the second tank is regulated by operating the switch controlling the flow of water to tank 1. The red dot's show the level to which we want to regulate the water. We assume the switch is operated by a motor and the operation of switching takes a minimum of 1 second and we take that to be our dwell time constraint.

Thus

$$J = \frac{1}{2}\int_0^{t_f} (x_2 - 3)^2 \, dt. \tag{63}$$

The switched system is initialized to $x_0 = (4, 2)'$ and optimized over time interval $[0, 20]$. We assume both the modes have the same minimum dwell time. The Armijo parameters are set to $\alpha = 0.5$ and $\beta = 0.5$. The time step for simulation was set to $dt = 1e^{-3}$ s. The mode sequence is initialized to $\sigma_0 = \{f_1\}$ for which the cost is $J = 11.23$. The algorithm is terminated when $|\theta_{\mathcal{X}}(\cdot)|$ falls below $\epsilon = 0.1e^{-3}$. The simulation is performed for dwell time constraints of 1s and the results are shown in Fig. 31 and the final results agree with our intuition.

In this chapter we assumed that each mode has the same dwell time, but in practice every mode can have its own minimum dwell time. This and other generalization of the dwell time constraints will be considered in the next chapter where we also consider the application of our results to solve the problem of optimal pesticide scheduling in precision agriculture.

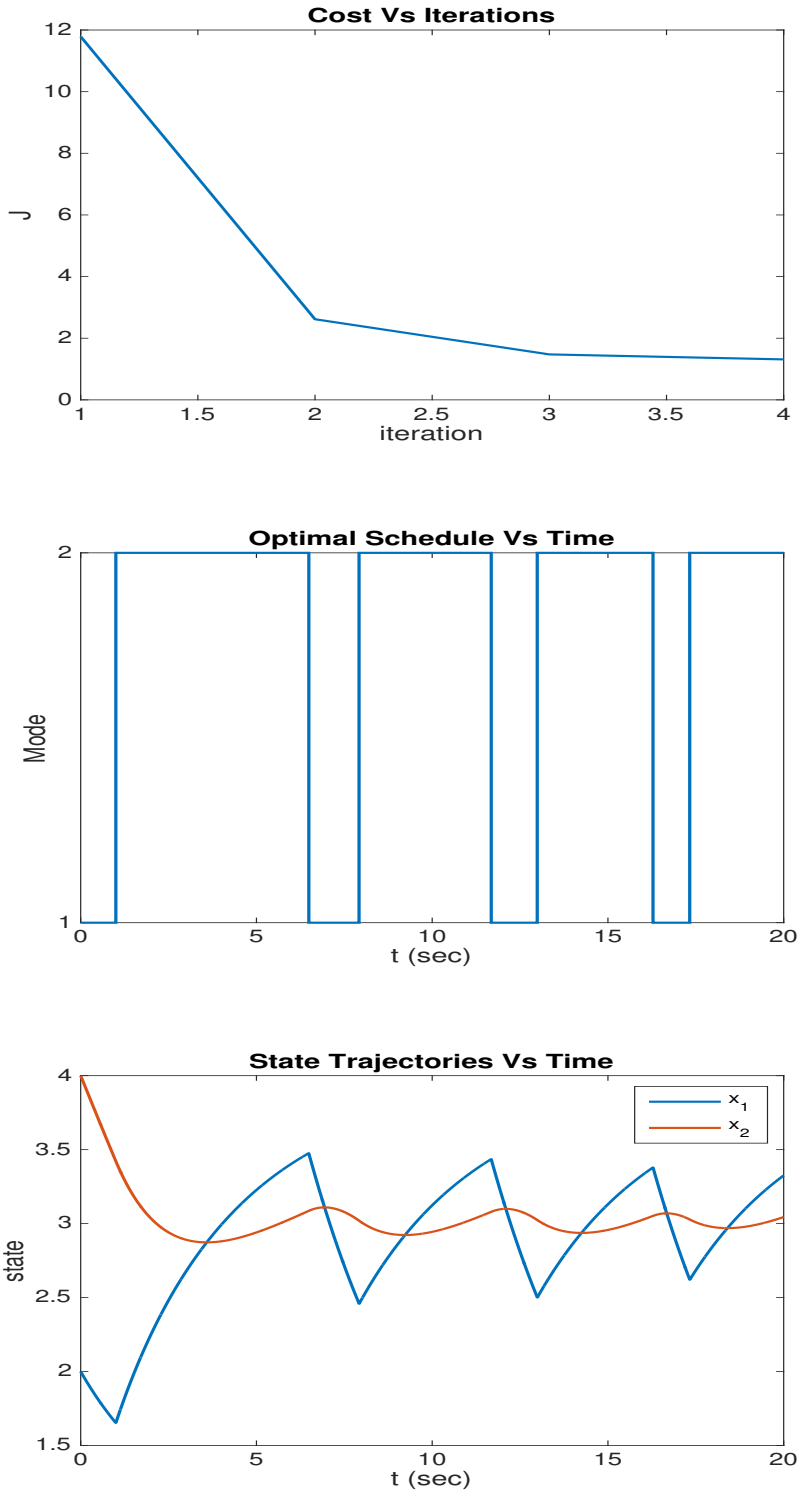


Figure 31: Plot of cost versus iterations, optimal mode schedule and state trajectories for $\delta = 1s$. The system wants to switch to mode 2 as soon as possible due to the lower level of water in tank 1 and then stays there for larger duration before it makes a switch to mode 1. The level of water in tank 2 is maintained close to the required level of 3.

CHAPTER V

THE DWELL TIME PROBLEM UNDER MODE SWITCHING CONSTRAINTS

The purpose of this chapter is twofolds. First we consider a generalization of the dwell time constraints to more general constraints on the sequence and next we consider the application of our framework to an interesting application in a diverse area of precision agriculture. Since any physical system that switches between different modes exhibits delay in switching and thus are amenable to the framework presented in the previous chapters, we particularly consider the problem of optimal pesticide scheduling to demonstrate the generality of our results, a work that also appeared in [45]. Before we can do that however, as is the case with any practical application, some extensions need to be made to the general theory to fit the application and this is what we do next. Although the extension presented below is inspired by the problem in precision agriculture, the extension again is general in nature and can be applied to any similar problems.

In this chapter, we consider the problem of minimizing a cost functional defined over the state trajectory as a function of switching times between different subsystems (modes), subject to various constraints on the mode switching and mode sequence. We generalize our results for the switch time optimization part of the dwell time problem in the following three ways. First, each mode can have its own minimum dwell time. Second, some of the modes can exist for fixed duration of time in the sequence resulting in equality constraints. Lastly, we consider constraints on non-adjacent switching times in the sequence. We present a gradient projection algorithm to solve the constrained hybrid optimal control problem and as a motivating example,

consider the application of our general results to a problem in precision agriculture, namely, the problem of optimal pesticide spray scheduling subject to various regulatory requirements and present a case study to demonstrate the generality of our results and utility of our approach.

As a motivating application, we investigate the problem of scheduling pesticide applications for commercially grown blueberries in our proposed framework. Since chemical substances used in pesticides may cause adverse effects in non-target species, their use is heavily regulated by the Environmental Protection Agency (EPA) in the United States [73] under the Federal Insecticide, Fungicide and Rodenticide Act [74]. In order to prevent dangerous substance interactions, some pesticides may not be applied within a certain time period of each other. Additionally, some pesticides cannot be applied within a certain amount of time before a planned harvest. Pesticide application is further governed by guidelines for effective use set by agricultural experts. For example, if a certain insecticide is applied once, it must be applied again at semi-regular intervals in order to prevent infestation of a spray-resistant population. To minimize the risk of infestation while ensuring responsible pesticide use, we formulate the problem as an optimal control problem. The evolution of risk of different pests is modeled as a stochastic switched dynamical system [23] that can switch between different behaviors (modes) depending on the pesticide being applied. We are interested in optimally selecting when to switch between these modes (equivalently, when to apply pesticides) so as to minimize a risk-dependent cost functional. Our results are demonstrated via simulation in which the constraints are based on expert recommendations.

The rest of the work presented in this chapter is organized as follows. In section 1, we consider the problem of general switching constraints and we present methods for solving it in section 2. In section 3, we give an overview of the problem in precision agriculture and consider the application of our results to solve a particular problem of

optimal pesticide scheduling for blueberry farming in section 4. Section 5 concludes the chapter with directions for future research.

5.1 Generalization of the Dwell Time Constraints

In the dwell time problem discussed previously, we made the assumption that every mode has the same dwell time and this resulted in constraints on the adjacent switching times such that the duration of each mode has to be greater than or equal to the minimum dwell time. In this chapter we extend the dwell time constrains in the following three ways:

1. Each mode can have its own minimum dwell time.

2. Constraints on the sequence of modes e.g mode 3 can occur only after mode 2.

3. Consider modes with fixed dwell times in the sequence i.e equality constraints.

These different type of constraints are illustrated in Fig. 32 as an example. We next give the problem definition for our switch time optimization problem subject to the above general constraints on the switching times.

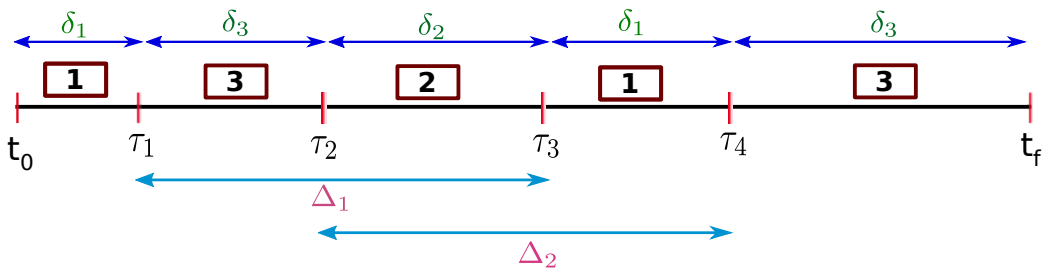


Figure 32: An example of hybrid system with 3 modes each having its own minimum dwell time δ_i. For a mode of fixed duration in the sequence, the corresponding δ_i denotes the equality constraints between the two adjacent switching times. The other constraints on the sequence are denoted by Δ_i.

Consider a set of subsystems $\Phi = \{f_q : q \in \mathcal{Q} = \{1, 2, \ldots, Q\}\}$ where each f_q is continuously differentiable. Suppose the mode sequence σ_0 is fixed with length $n+1$, so that the dynamics of the autonomous system can be represented as

$$\begin{aligned} &\dot{x}(t) = f_{\sigma_0(i)}(x(t)), \quad t \in [\tau_{i-1}, \tau_i), \quad i = 1, \ldots, n+1, \\ &x(0) = x_0, \end{aligned} \tag{64}$$

where $\tau = (\tau_0, \tau_1, \ldots, \tau_{n+1})^\top$ is the vector of transition times between the different modes with $\tau_0 = t_0$ and $\tau_{n+1} = t_f$, and x_0 denotes the initial condition. The representation used here is commonly referred to as mode schedule representation of hybrid switched systems and while other representation exists useful in other contexts, there exists a bijection between the different representations [5]. We will stick to the mode schedule representation since it best serves our purpose here.

The constraints on the mode switchings results in an optimization space that is a subset of $\mathbb{R}^{n+1}$ and we next proceed to define this optimization space explicitly in terms of the three constraints introduced in the introduction. In what follows next, we use δ to denote the adjacent constraints and Δ to indicate non-adjacent constraints on the switching times. The minimum dwell time constraint on each mode translates to the constraint

$$a_i^\top \tau >= \delta_i \tag{65}$$

on the switching times, where $a_i = (0, \ldots, 0, -1, 1, 0, \ldots, 0)^\top$ for $i \in W \subset \{1, \ldots, n\}$ with the non-zero terms at the locations $i-1$ and i respectively and δ_i is the mode-dependent dwell time. Moreover, when $\sigma_0(i) = \sigma_0(j)$, we have $\delta_i = \delta_j$, i.e. the same modes have the same dwell time and we assume that we have k such constraints. Suppose we have l constraints corresponding to the constraints on the sequence. These constraints can be represented as

$$b_i^\top \tau >= \Delta_i \tag{66}$$

for some b_i, $i = 1, \ldots, l$. Finally, some of the modes in the sequence are of fixed duration or fixed dwell time. Suppose we have m such constraints, then these equality constraints can be written as

$$c_i^\top \tau_i = \delta_i \tag{67}$$

for some c_i, $i = 1, \ldots, m$. The inequality constraints (65) and (66) can be cascaded together so we get the inequality constraints in the matrix form

$$A_{ie}\tau >= D_{ie} \tag{68}$$

where $A_{ie} \in \mathbb{R}^{(k+l)\times(n+1)}$ and $D_{ie} \in \mathbb{R}^{k+l}$. Similarly we can write the equality constraints (67) in matrix form as

$$A_{eq}\tau = D_{eq} \tag{69}$$

with $A_{eq} \in \mathbb{R}^{(m)\times(n+1)}$ and $D_{eq} \in \mathbb{R}^m$. With this, we can defined the switch time optimization space as

$$\mathcal{T} = \left\{\tau \in \mathbb{R}^{n+1} \quad | \quad A_{ie}\tau >= D_{ie}, \quad A_{eq}\tau = D_{eq}\right\} \tag{70}$$

We next give the problem definition for our switch time optimization problem subject to the above general constraints on the switching times.

Hybrid Optimal Control Problem

Let $L : \mathbb{R}^n \to R$ be a continuously differentiable function and consider the cost functional

$$J = \int_0^{t_f} L(x(t))\, dt + \phi(x(t_f)). \tag{71}$$

The optimal control problem that we want to solve in this chapter then is

$$\min_{\tau\in\mathcal{T}} \quad J(\tau) \tag{72}$$

subject to the dynamical and initial condition constraints (64).

5.2 Computing the Optimal Controls

Solving the optimal control problems analytically in general is hard and one has to resort to some sort of numerical scheme to compute the optimal controls. In this section, we present an algorithm to solve the constrained hybrid optimal control problem presented in previous section. Essential to any algorithm are the descent direction and the step size in the direction of descent. In the absence of any constraints on switching, an expression for the gradient of the cost with respect to switching times was derived in [26] using variational methods where each component of the gradient vector ∇J is given by

$$\frac{dJ}{d\tau_i} = p(\tau_i)\left(f_{\sigma_0(i)}(x(\tau_i)) - f_{\sigma_0(i+1)}(x(\tau_i))\right) \tag{73}$$

and where $x(t)$ is the solution of the system (64) and $p(t)$ is solution of the costate equation

$$\dot{p}(t) = -p(t)\frac{\partial f_{\sigma_0}(x(t))}{\partial x} - \frac{\partial L(x(t))}{\partial x}, \quad p(t_f) = \frac{\partial \phi(x(t_f))}{\partial x}. \tag{74}$$

The negative of this gradient gives then the direction of steepest descent which in conjunction with Armijo step-size rule [9] is used to solve the switch time optimization problem.

When the constraint set is convex and compact, the feasible descent direction can be found by projecting the update vector obtained using steepest descent onto the constraint set. This requires solving a quadratic program in general. However, when the constraint set has a special structure, such as that of a box or polyhedron, one can use manifold suboptimization [62]. This is a type of gradient projection method where instead of projecting the update vector onto the entire constraint set, the gradient is projected onto a linear manifold of active constraints. This greatly simplifies the computation of projection [13]. Our constraint set is still convex and compact and has the structure of polyhedron, which allows for easy computation of projection onto

the constraint set and hence the use of manifold suboptimization method for solving the problem as described below.

Identify the set of active constraints in (68) namely

$$\mathcal{I} = \left\{ i \in \mathbb{Z}^+ \quad | \quad A^i_{ie}\tau = D^i_{ie} \right\}, \tag{75}$$

where the superscript i denotes the row for which the equality holds. Suppose a subset $q <= (k+l)$ of these constraints in (68) are active and let

$$A^q_{ie}\tau = D^q_{ie}, \tag{76}$$

be the corresponding matrix representation of the active constraints at a feasible τ. We concatenate these constraints with the m equality constraints in (69), which effectively can be viewed as *always active* constraints. We can write down the equality constraints (69) and (76) in compact form as

$$N\,\tau = B, \qquad N = \begin{bmatrix} A^q_{ie} \\ \dots \\ A_{eq} \end{bmatrix}, \quad B = \begin{bmatrix} D^q_{ie} \\ \dots \\ D_{eq} \end{bmatrix}. \tag{77}$$

The feasible descent direction is then given by

$$d = -P\,\nabla J(\tau), \tag{78}$$

where

$$P = I - N^T(NN^T)^{-1}N \tag{79}$$

is the projection and N is the matrix of equality constraints in equation (77). The Karush-Kuhn-Tucker (KKT) multiplier associated with the quadratic program is given by

$$\mu = -(NN^T)^{-1}N^T\nabla J(\tau). \tag{80}$$

where $\mu \in \mathbb{R}^{q+m}$, with q of the multipliers associated with the active constraints and the remaining m associated with the *always active* equality constraints.

From KKT necessary conditions for optimality, q of the multipliers corresponding to the active constraints in the set of inequality constraints need to be non-negative while the the remaining m constraints can take any value. During the manifold sub-optimization, if some of the q multipliers are negative, the active constraint associated with the largest negative value is usually dropped and the optimization process is repeated, till either all the q multipliers associated with the active constraints become greater than or equal to zero.

To ensure that the algorithm converges, the step size in the descent direction d is obtained by using Armijo step size rule [9] over the set

$$\Lambda = \{\lambda > 0 \quad | \quad A^j_{ie}(\tau + \lambda d) \geq D^j_{ie}, \quad j \notin \mathcal{I}\},$$

where the superscript j denotes the row corresponding to the inactive constraint. From this set, we can compute the upper limit on λ as

$$\lambda_{max} = \min_{j \notin \mathcal{I}} \frac{A^j_{ie}\tau - D^j_{ie}}{A^j_{ie}d}, \tag{81}$$

such that $A^j_{ie}d < 0$, otherwise there is no upper limit due to the j^{th} inactive constraint on λ. In case there is no upper limit on λ, we set $\lambda = \bar{\lambda}$ for some fixed $\bar{\lambda}$. So we have $\lambda = \gamma\lambda_{max}$ where $\gamma \in (0, 1]$ to allow for step size in the range$(0, \lambda_{max}]$. We let $d_s = \lambda_{max}d$ denote the scaled descent direction, where d is as defined in Eq. (79). The next iteration can then be written as

$$\tau_{next} = \tau + \gamma d_s. \tag{82}$$

where the step size γ is chosen according to Armijo rule [9] for the algorithm to converge to a stationary point as follows. Let $\alpha \in (0, 1]$, $\beta \in (0, 1)$ be constants, then the step size in the descent direction is chosen as

$$\gamma = \beta^r \quad : \quad r = \min_{j \in \mathbb{Z}^+} \left\{ J(\tau + \beta^j d_s) - J(\tau) \leq \beta^j \alpha \left\langle \nabla J(\tau), d_s \right\rangle \right\} \tag{83}$$

where $\langle ., . \rangle$ denotes the standard inner product and the term on the right hand side of the inequality is always negative or zero. The Armijo rule ensures that the cost

at the next iteration is always less than the cost at the current iteration by some definite amount, thus ascribing the algorithm with the property of sufficient descent and hence convergence of our algorithm. See [57] for discussion on the property of sufficient descent and algorithm convergence.

With the descent direction and step size available, we next state the algorithm for solving our switch time optimization problem.

GRADIENT PROJECTION ALGORITHM

Choose $\alpha \in (0, 1]$, $\beta \in (0, 1)$, $\bar{\lambda}$ and $\tau \in \mathcal{T}$. Identify the set of active constraints $\mathcal{I}$ (75) and repeat.

1. Find the descent direction d using (78).
2. If $d = 0$, compute μ using (80). If q entries of μ are all non-negative, **Stop**. If not, eliminate the active constraint associated with the most negative μ from the q entries and go to step 1.
3. Compute the step size γ using (81) and (83).
4. Update the switch time vector τ using (82) and update the set of active constraints (77). Go to step 1.

The algorithm converges to a stationary point [57] and the Armijo step size rule helps the algorithm converge to the optimal solution during the first few iterations as will be demonstrated next in an illustrative example, and then in application to optimal pesticide scheduling.

5.2.1 Example

We consider the *'Two Switched System'* of chapter 3. The matrices associated with the dynamics of two linear subsystems are

$$A_1 = \begin{pmatrix} -1 & 0 \\ 1 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 1 \\ 1 & -2 \end{pmatrix}$$

and a quadratic cost functional on the state trajectory defined as

$$J = \frac{1}{2}\int_0^T ||x(t)||^2 dt.$$

Suppose the dwell time constraint associated with mode 1 is $\delta_1 = 0.1$ s and that with mode 2 is $\delta = 0.2$s. We consider the sequence $\sigma_0 = \{1, 2, 1, 2, 1\}$ and assume that mode 2 in this sequence has to exist for a fixed duration of 0.2s i.e an equality constraint, while the rest of the modes in the sequence have to satisfy their respective minimum dwell time constraints leading to inequality constraints. We introduce a constraint on the sequence that the mode 2 cannot occur in the sequence before 0.5 seconds after it has occurred the first time, resulting in another inequality constraint. We start with a feasible τ and solve the problem over the horizon $t_f = 2$ seconds. The Armijo parameters are set to $\alpha = 0.01$ and $\beta = 0.5$ and the differential equations are solved using a step size of $dt = 0.0001$ s. The results are shown in Fig. 33 - 36 and we next briefly explain them.

Fig. 33 shows the cost vs. iterations plot. The algorithm converges in about 82 iterations and the cost reduces from the initial value of 10.81 to the final value of 2.76. The sudden decrease in cost occurs as the active constraints corresponding to most negative q KKT multipliers are dropped from the set $\mathcal{I}$. The staircase appearance indicates rapid convergence at each manifold suboptimization step, thanks to Armijo step size rule.

Fig. 34 shows the mode insertion gradients at the end of switch time optimization. As expected, the insertion gradients at the optimal switching times for the inactive

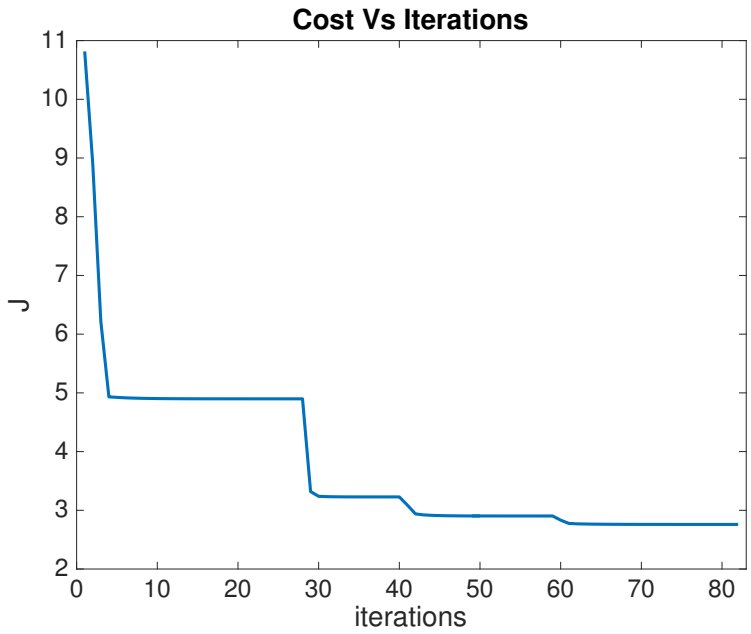


Figure 33: The cost shows a rapid decrease during the first few iterations and then the next time constraint is dropped, there is again a rapid decrease in cost, resulting in almost a staircase like shape. The process repeats until optimal conditions are satisfied.

constraints are zero, indicating optimal location of the switching times . For the mode with equality constraints, the insertion gradients at the two switching times are equal in value, instead of being zero, since the switch times are not free to move.

The optimal switching times between different modes subject to the above mentioned constraints are shown in Fig. 35. The switching times, which were initialized to $\tau = (0,\ 0.2,\ 0.5,\ 0.9,\ 1.1,\ 2)^\top$ are optimized to $\tau_{opt} = (0,\ 0.74,\ 1.32,\ 1.57,\ 1.77,\ 2)^\top$. Note in particular, how the fourth mode in the sequence, i.e mode 2 with fixed dwell time of 0.2s has moved. The evolution of states as a function of time for the optimal switching times is shown in Fig. 36 where the abrupt changes in state trajectory correspond to the transition times in Fig. 35(a). Our work on the hybrid optimal control under mode switching constraints has been submitted to [3].

Next we consider the application of the dwell time problem with general constraints to an interesting problem in precision agriculture i.e the problem of optimal

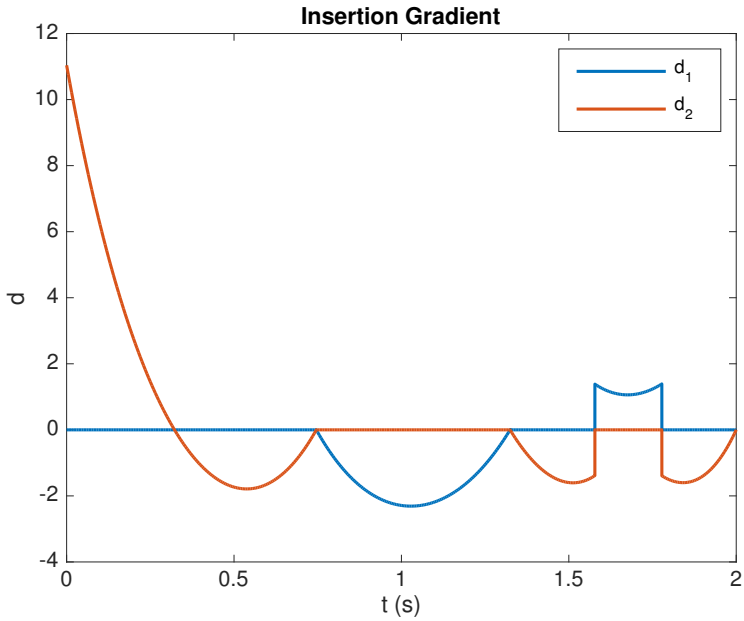


Figure 34: The insertion gradients are zero at the optimal switching times except for the mode that has to exist for duration of fixed time in the sequence. At its optimal position, the insertion gradient values at the corresponding switching times are equal.

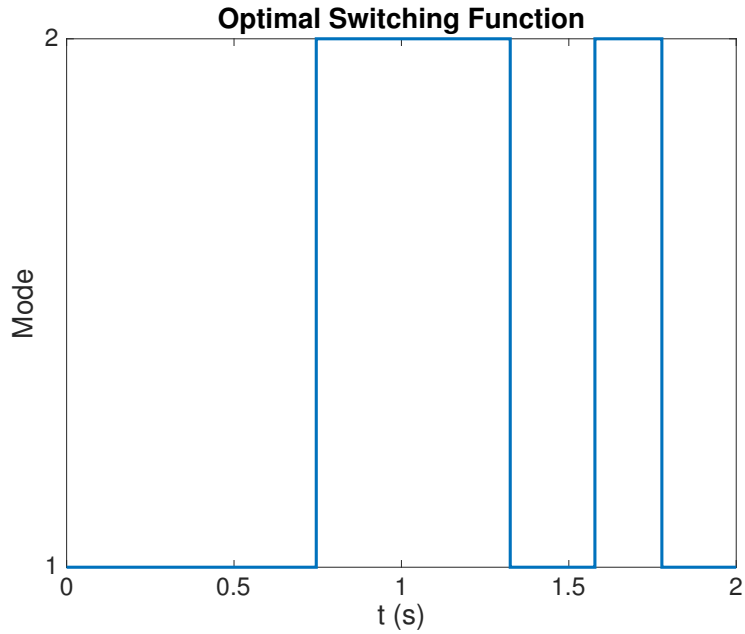


Figure 35: Switching between the two modes . Note that the fourth element in the sequence, namely, mode 2 exists for the duration of dwell time and is thus always an active constraint. The switching function respects all the switch time constraints.

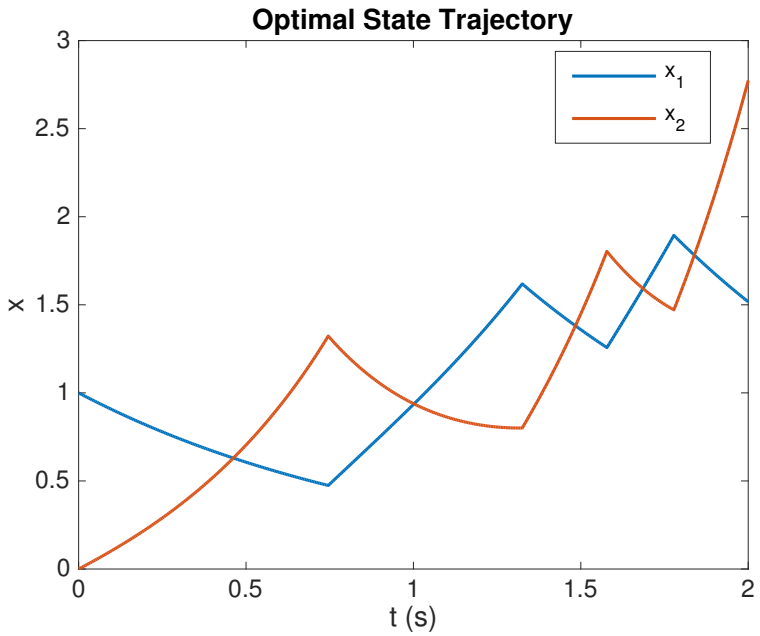


Figure 36: Evolution of state trajectories as a function of time. The abrupt changes in state correspond to the the switching instances in Fig. 35.

pesticide scheduling. This work was published in [45] and the highlights of this work are reproduced in the following section, starting with the motivation for studying this problem.

5.3 Application - Optimal Pesticide Scheduling

Food crop cultivation is a high-risk proposition. Crop yields can be dramatically affected by external events, such as sudden frosts, fungal infections, or variations in soil nitrogen content. Many yield improvement tools developed in recent years, such as genetic modification and soil nitrogenation, are chemical and biological in nature. Recent droughts in the state of California, which are responsible for $2.2 billion USD in losses [42], have demonstrated that yield optimization requires effective resource management in addition to these new biotechnologies [31]. The emerging field of precision agriculture (PA) intends to provide tools, such as distributed plant health sensors, to enable growers to make informed decisions about how best to use their

resources

The rest of the work in this section is organized as follows. First we present an overview of pesticide scheduling for commercial blueberry crops. Next we give the problem definition followed by setting up the problem as switch time optimization problem subject to the generalized dwell time constraints presented in the previous section. Finally, we present a simulation case study using 4 pesticides and Section 6 concludes the chapter and presents some open problems in precision agriculture.

5.3.1 Pesticide Scheduling for Commercial Blueberry Crops

In this section, we consider optimal pesticide scheduling for blueberry crops as an application of our constrained hybrid optimal control framework presented in section-2. This problem exhibits many of the features that makes precision agriculture interesting: discrete decisions that must be made to ensure system health, seasonally-dependent goals and constraints, and external disturbances. Though we focus on blueberries, this problem formulation applies to many applications in agriculture in which actions must be critically timed in order to ensure healthy crops. Our framework can directly be applied to cultivation other commercially-grown fruit such as grapes or raspberries.

We use the term pesticide to include any number of insecticides, fungicides, or rodenticide. The term "blueberry" refers to any member of the genus *Vaccinium* native to North and South America[1]. The most widely grown commercial variety is *Vaccinium corymbosum*, or the highbush blueberry. Blueberries are flowering perennial bushes that flower in the spring (February to March) and are harvested in the summer (June to August).

In order to prescribe preventative pest management measures, agricultural experts

[1]Members of genus *Vaccinium* native to Europe are called "bilberries"

construct data-driven models for disease progression and risk overtime. For blueberries, some models already exist for Septoria leaf spot [53] and mummy berry [64]. Inputs to these models include factors such as observations of disease, ambient temperature, and humidity levels. Currently, pesticide spraying schedules are determined on ad-hoc basis by farmers with input from regional agricultural experts. This process is susceptible to error and onerous to growers. In this chapter, we propose an automatic scheduling algorithm that reduces expected pest risk and respects constraints on pesticide use.

5.3.2 Pesticide Regulation

Pesticides are dissolved in water and dispersed throughout the field via "airblast" sprayers. At the beginning of the growing season, region-specific pesticide spray guides developed by crop specialists are issued to growers, [47]. The four main types of constraints indicated by these guides are the following.

1. **Pre-Harvest Interval (PHI)** Each pesticide has a defined Pre-Harvest Interval (PHI), which is the amount of time after a pesticide has been sprayed that must elapse before plants may be harvested. For Spinosad spray, this is four days [47].

2. **Lifecycle Requirements** Constraints on application times are determined by the stages of the plants' lifecycle. For example, Horticultural Oil can only be used before the bloom phase [47].

3. **Substance Interactions** Certain pesticides, when used concurrently can produce adverse effects to plants, wildlife, or humans. For example, Lime Sulfur and Horticultural Oil cannot be used within 14 days of each other [47].

4. **Repeated Applications** Repeated applications at regular intervals may be required to prevent pesticide-resistant pest populations. Conversely, in order to

limit the secondary effects of some substances, the number of repeated sprayings may be regulated. For example, Spinosad bait must be applied every 7 days [47]

5.3.3 Pesticide Spray Problem as a Hybrid Optimal Control Problem

In this section, we formalize what we mean by a pesticide schedule and formalize the constraints presented in the previous section. In our approach, spraying constraints 1-4 are handled by encoding them as linear equality and inequality constraints on switching times. We explicitly model the risk of infection and minimize it using optimal control.

We consider pesticide application over the time interval $[0, t_f]$ where 0 corresponds to the beginning of the growth season and t_f corresponds to the harvest time of the system. Times are given in units of days.

5.3.4 Pest Risk Evolution

Pest control measures are scheduled by farmers throughout the growing season as the presence of particular threats becomes more or less likely. In this section, we consider a problem formulation that "closes the loop" between pest risk models derived by experts and the pesticide scheduling problem.

For a given collection of pesticides $P = \{P_j\}_{j=1}^{L}$, define the *risk state* of the crop as a vector $x = [x_k]_{k=1}^{M} \in [0,1]^M$, where x_k is the probability that pest k is present. Associate with each pest a *risk growth rate* $r_{g,k} > 0$ and a *pesticide effectiveness rate* $r_{e,k} < 0$.

Define the set of matrices $F = \{F_j\}_{j=0}^{M}$ such that

$$\begin{aligned} F_0 &= \text{diag}(r_{g,1}, \ldots, r_{g,L}) \\ F_j &= \text{diag}(r_{j,1}, \ldots r_{j,L}). \\ r_{j,k} &= \begin{cases} r_{e,j} & P_j \text{ prevents pest } k \\ r_{g,j} & \text{else} \end{cases} \end{aligned} \tag{84}$$

Given a static schedule S, define the induced *mode schedule* as a function $\sigma_S : [0, t_f] \to \{0, 1, \ldots, M\}$ where

$$\begin{aligned} &\sigma_S(t) = j \quad \exists \quad t' \in [t, t-\epsilon), \quad P_j \in x_s(t') \\ &\sigma_S(t) = 0 \quad \text{else} \end{aligned} \tag{85}$$

Then, the pest risk vector evolves over time $[0, t_f]$ according to the *switched stochastic system* [23]

$$\dot{x}(t) = F_{\sigma_S(t)}(x) + w(t) \tag{86}$$

where $w(t) = [w_j(t)]_{j=1}^{M}$ is a zero mean stationary Gaussian process with covariance

$$Q(t) = E[w(t)w(t)^T] = diag(q_1, q_2, \ldots, q_M). \tag{87}$$

Intuitively, when no pesticide is being applied, the risk vector evolves according to $\dot{x}(t) = F_0 x(t) + w(t)$. That is, the risk grows over time with an average rate $r_{g,j}$. The noise $w(t)$ represents the effects of uncertainties such as weather or disease observations that can cause the risk to increase or decrease. When a pesticide P_j is sprayed at time t, the risk vector evolves according to $\dot{x}(t') = F_j x(t') + w(t')$ $\forall t' \in [t, t+\epsilon)$. That is, the risk of the pests that are treated by P_j are on average reducing with rates $r_{e,k}$. This reflects the reduction in infestation risk due to pesticide application. We make the assumption that $-r_{e,k} >> r_{g,k}$.

5.3.5 Mode Sequence Selection

In order to reduce the search space of schedules, we fix the sequence of pesticide sprays according to a risk-based heuristic. Let

$$Ind(P_j) = \{k \quad | \quad P^k = P_j\}. \tag{88}$$

To fix the mode sequence, we fix a risk threshold x_{thresh} and compute the unsprayed threshold crossing time vector as

$$t_{thresh} = [t_{thresh,k}]_{k=1}^{N} \left[\ln \left(\frac{x_{thresh} - x_0}{r_{g,k}} \right) \right]_{k=1}^{M} \tag{89}$$

where $r_{g,j}, x_0$ are as defined in Section 5.3.4. Then, the pesticides we actually spray are

$$P_{sprayed} = \{j \quad | \quad t_{thresh,k} \leq t_f, \quad P_j \text{ treats pest } k\} \tag{90}$$

and the mode sequence $Seq = P^1 P^2 \dots P^{|P_{sprayed}|}$ is defined such that pesticide P^m treats x_k where $t_{thresh,k}$ is the m^{th} smallest value of t_{thresh}.

Fig. 37 shows an example of switch times $\tau_S = [\tau_1, \tau_2, \tau_3, \tau_4]$ for a fixed mode sequence $Seq = \{P_1, P_2, P_1, P_2, P_1\}$.

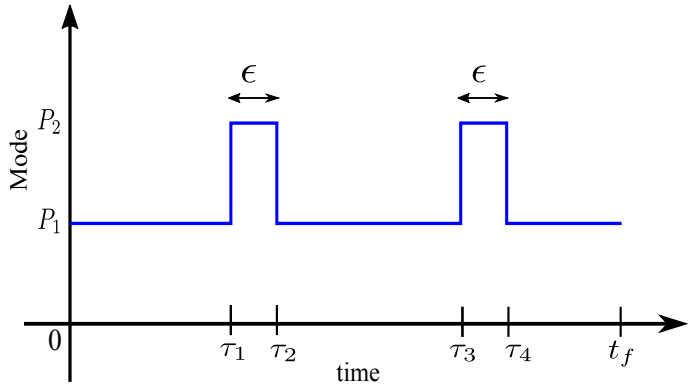


Figure 37: Example of switch times for a fixed mode sequence satisfying switch time constraints. The fixed durations of ϵ are when pesticides are sprayed and they can be different for different pesticides. They correspond to the equality constraints in our discussion in previous sections.

Table 1: Translating constraints on pesticides spray to linear constraints on switch time vectors. The different constraints are captured using the three types of constraints on mode switchings mentioned in the previous section.

Requirement	**Linear Constraint**
Pre-Harvest Interval	$\tau_{2k} \leq t_f - PHI_j$ $\forall k \in Ind(P_j)$
Lifecycle Requirements	$stage_{LC,s} \leq \tau_{2k-1} \leq stage_{LC,e}$ $\forall k \in Ind(P_j)$
Substance Interactions	$\tau_{2k-1} \geq \tau_{2p-1} + overlap_{j,m}$ $\forall k \in Ind(P_j), \forall p \in Ind(P_m)$
Repeated Interactions	$next_{j,1} \leq \tau_{2k-1} - \tau_{2m-1} \leq next_{j,2}$ $Ind(P_j) = \{k, m\}, k < m$

Now, define the matrix A_{onoff} as

$$A_{onoff} \quad = \left[A_{onoff,i,j}\right]_{i=1,j=1}^{N,N},$$

where

$$A_{onoff,i,j} = \begin{cases} 1, & i = j, \\ -1, & j = i+1, \\ 0, & \text{else.} \end{cases} \tag{91}$$

We can now formally state the optimal pesticide scheduling problem as

Problem 1 ***Linear Constraints*** *Minimize*

$$\begin{gathered} J(\tau) = E\left[\int_0^{t_f} x(t)^T C x(t) dt\right] \\ \textit{subject to} \\ \dot{x} = F_{\sigma_0(t))} x(t) + w(t) \\ x(0) = x_0 \\ A_{onoff}\, \tau_S = \epsilon \mathbf{1}_N \\ A_{spray}\, \tau_S \leq b_{spray} \end{gathered} \tag{92}$$

where $C \in \mathbb{R}^{M \times M}$.

Before we employ our framework, we need information about the cost gradient with respect to switching times in the absence of constraints. For stochastic systems, it has been shown in [48] that the derivative of the cost with respect to switching times for this type of problem is given as

$$\frac{dJ}{d\tau_j} = \frac{1}{2}\sum_{i=j}^{N}\int_{\tau_i}^{\tau_{i+1}} Tr\left(e^{F_{i+1}(t-\tau_i)}\Gamma_{ji}e^{F_{i+1}^T(t-\tau_i)}C\right)dt \tag{93}$$

where

$$\begin{aligned}
&\Gamma_{jj} = M_j, \quad j = 1,\ldots,N,\\
&\Gamma_{ji} = e^{F_i(\tau_i-\tau_{i-1})}\Gamma_{j,i-1}e^{F_i^T(\tau_i-\tau_{i-1})}, \quad i = j+1,\ldots,N,\\
&M_j = (F_j - F_{j+1})m_x(\tau_j) + m_x(\tau_j)(F_j^T - F_{j+1}^T),\ j = 1,\ldots,N
\end{aligned}$$

and $m_x(t)$ is the second moment of x and is computed via the matrix Riccati equation

$$\begin{aligned}
\dot{m}_x(t) &= F_i m_x(t) + m_x(t)F_i^T + Q,\ t \in (\tau_{i-1},\tau_i],\\
m_x(t_0) &= P0 + m_0 m_0^T,
\end{aligned}$$

for $i = 1,\ldots,N$. Here $m_0 = E[x(t_0)]$ is the mean and $P_0 = E[(x(t_0)-x_0)(x(t_0)-x_0)^T]$ is the variance in initial condition of state $x(t)$. With the information on gradient and other quantities available, we use the gradient projection algorithm of section 5.2 to solve the problem of optimal pesticide scheduling in the next section.

5.4 Case Study

To demonstrate the implementation of the proposed switch time optimization of the pesticide scheduling problem, we consider a case study involving four pesticides that treat six different pests as listed in Table 2. Parameters for each of these pesticides were derived from the spray guide [47]. The time bounds for constraints 1, 3, and 4

are all given explicitly in this guide. For constraint 2, i.e., the life stage requirements, we assumed a 120 day growing season and demarcated the life stages appropriately, e.g., 30 days dormant, 20 days budding, 20 days blooming, 30 days ripening, and 20 days harvest. A total of 38 linear constraints were derived from the guide. The initial risks and growth rates for the risk were chosen such that x_{thresh} was met some time during the allowed life stage for each pest. Effectiveness of each pesticide/pest pair in the guide is rated on a scale of one to four. The effectiveness rates used in our model scale with this number. The noise matrix was non-uniformly weighted with the largest diagonal entry being four times as large as the smallest entry. The cost matrix was also non-uniformly weighted with values in the range $1 - 10$.

Table 2: List of pesticides and pests considered in case study. Four pesticides are selected to treat six pests, so some pesticides can treat more than one pests. The constraints on the use of these different pesticides are taken from spray guide.

Pesticide	**Pests Treated**
Lime Sulfur (L.S.)	*Exobasidium*
Horticultural Oil (H.O.)	Scale Blueberry Bud Mite
Bacillus thuringenesis (B.t.)	Fruit Worm Red Humped Caterpillar
Pyrethrins (Py)	Spotted Wing Drosophila

The results of our Matlab simulation are shown in Fig. 38. This simulation took 105s on a computer with 2.7GHz 2-core processor with 8GB RAM. We can see from Fig. 38(a) that the scheduling algorithm requires very few iterations to converge. In Figure 38(b), we see the optimal mode schedule that is the result of this algorithm. The height of each spike corresponds to which pesticide is being applied (shown on the y axis). In addition to seasonal constraints and PHI constraints, the optimal mode schedule satisifies the following constraints

1. Horticultural Oil and Lime Sulfur cannot be applied within 14 days of each other.

2. If Pyrethrins is sprayed once, it must be sprayed again "weekly", which we interpreted as "within the next 4 to 10 days".

Fig. 38(c) and 38(d) shows the evolution of the risk trajectories under an initial feasible schedule and the optimal schedule, respectively. In most cases, spray times in the optimal schedule are later than in the initial schedule. We posit that this is due to the effect of noise in the model,i.e., it is better to minimize risk later in the model rather than use our resources too early and risk unacceptable risk growth later. We also note that the intervals between applications of the same pesticides is longer in the optimal case than in the initial schedule. This can be seen in the trajectory of the risk for SWD. This policy makes sense, as we would expect that in order to be most effective, it is better to apply pesticides at semi-regular intervals. That is, it is better to space out repeated applications of pesticides in order to control for potential future increases in risk due to the external events modeled by the noise.

5.5 Concluding Remarks

In this chapter, we extended the results of our previous two chapters by considering more general constraints on the mode switches. While we considered the application to a specific problem in precision agriculture, the framework presentee in this chapter and the previous chapters is quite general and since any practical switched system or process is subject to some kind of dwell time constraints, the results presented in chapter 3-5 are of great practical importance and are expected to find widespread applications because of the generality of results.

In this chapter, considered the hybrid optimal control problem with a number of constraints on the mode switchings and presented a gradient projection based algorithm to compute the optimal controls. As an application of our general results, we considered the problem of optimal pesticide scheduling in precision agriculture, while highlighting the significance of the application problem considered. We formulated

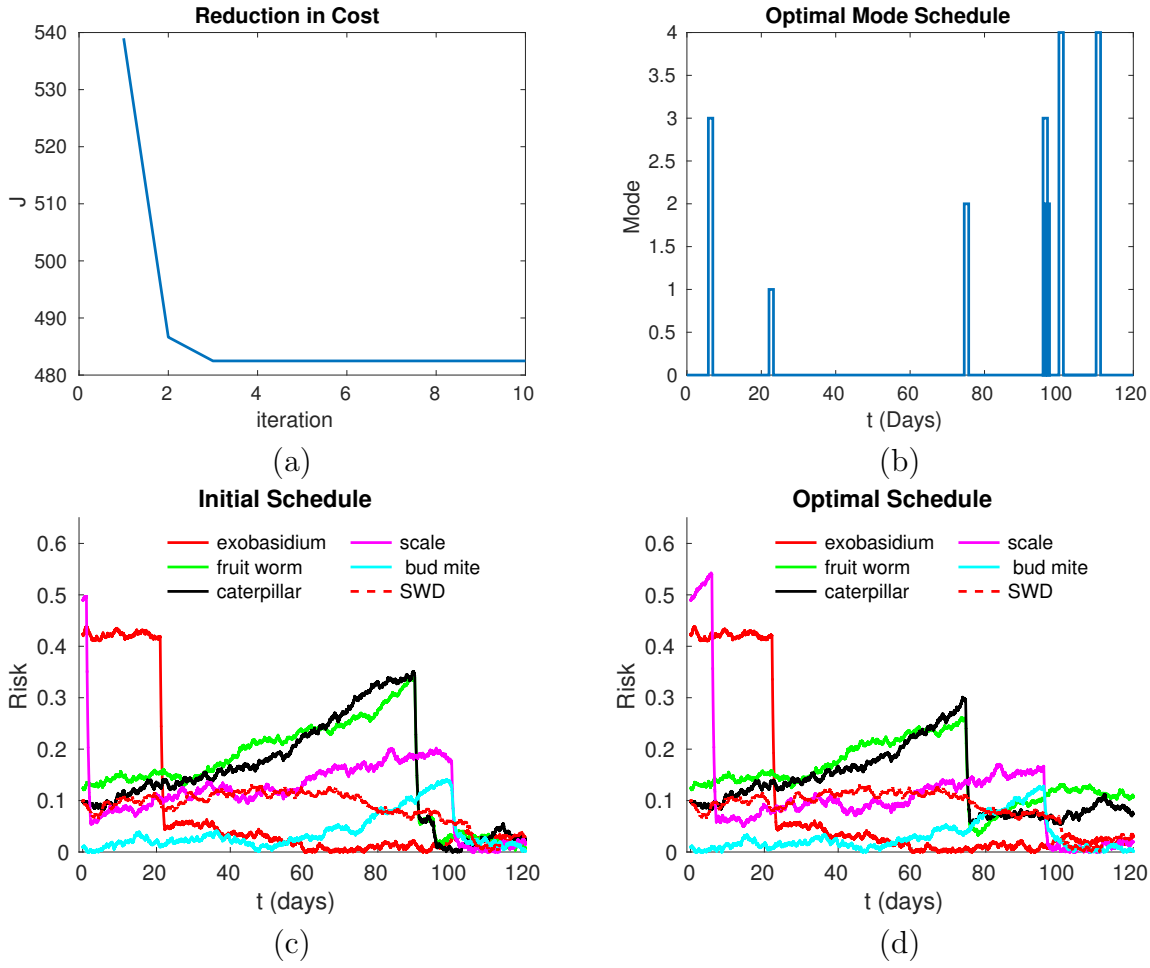


Figure 38: Results of case study. The subplots show (a)Reduction in cost. (b) Optimal mode schedule. (c) Initial risk trajectories. (d) Risk trajectories under Optimal Schedule.

the problem in the framework of constrained hybrid optimal control and presented a case study for the application of our general results. Future improvements to this solution include implementing receding horizon scheduling in which the optimal schedule is recomputed daily based on the current rather than predicted risk of pesticides. Our results indicate the existence of relevant problems in agriculture that can be addressed with systems theory. In the future, we propose to take such an interdisciplinary approach to address the problems of integrated pest management [30], smart irrigation [50], and persistent crop management [68].

While we considered the application to a specific problem in precision agriculture, the framework presented in this chapter and the previous chapters is quite general and since any practical switched system or process is subject to some kind of dwell time constraints, the results presented in chapter 2-5 are of great practical importance and are expected to find widespread applications because of the generality of results.

In the forthcoming chapters, we consider another set of constrained hybrid optimal control problem inspired by problems in the area of power aware mobile robotic networks, where the objective is to balance the motion and communication energy of robots in a network subject to various constraints on the input and state. We first solve the problem as an ordinary optimal control problem and then later present how the problem is hybrid not only in object but also in dynamics and solve it in the framework of constrained hybrid optimal control.

CHAPTER VI

MOTION AND COMMUNICATION CO-OPTIMIZATION IN FADING ENVIRONMENTS

The emerging field of communication-aware robotics is drawing from the established areas of sensor networks on the one hand, and mobile robotic networks on the other hand. A principal issue in mobile sensor networks concerns power management due to the limited energy available to each sensor, and an important question is how to schedule sensing and transmission in a way that saves energy and prolongs the network's deployment. Major issues in networked robotics concern coordination, control, and communications, including decentralized decision making and task coordination by networks of agents. In chapter 2, we provided a brief overview of the main problems and results in the area of power aware mobile robotic networks.

This problem of co-optimizing motion and communication energy in multi-agent robotics in fading environments has the structure of a hybrid optimal control problem and formulating this problem as an optimal control problem and subsequently solving them using efficient algorithms to achieve co-ordinated tasks is another main focus of this research, and hence this chapter. The control parameters for this problem are acceleration, which is continuous, and spectral efficiency which is a discrete and the problem is hybrid both in the objective function and dynamics with a number of constraints on the trajectory and controls. We solve the co-optimization problem first with the assumption that the spectral efficiency is a continuous variable in this and the next chapter and the problem is solved as a true switched hybrid dynamical system in chapter 8. The framework is developed in the general setting and application examples are provided to validate the results.

In this chapter, we consider the problem of motion and communication planning in power aware mobile robotic networks. The objective is to minimize the energy expanded in the performance of co-ordinated tasks such as transmitting data from a subset of the terrain to a remote station while in motion. The principal forms of the power and energy required for these tasks are due to transmission and motion. The channel quality is not uniform and it is assessed at unvisited locations based on few a priori measurements. The problem is posed in the framework of optimal control using realistic models for channel communication and motion power. We apply to it a specialized algorithm which approaches solution (minimum) points rapidly, thereby raising the possibility of its real-time implementation, which will be dealt with in the next chapter Simulation results suggest that the proposed approach can be effective in realistic applications.

Reference [84] is the starting point for the work presented in this chapter.. More specifically, a mobile robot needs to transmit a fixed number of information bits to a remote station in a realistic fading communication environment, while moving along a given trajectory. Our goal is to minimize the total energy cost of the robot, including both communication and motion costs, while satisfying time budget and communication reception quality constraints. In [84], this problem is solved in a discrete fashion where a first-order dynamical model for the robot is assumed, without considering the acceleration cost. In practice, it is important to consider the acceleration cost. For instance, if the channel quality is improving, the robot may need to slow down to take advantage of the good connectivity for transmission. However, a realistic motion model that considers the acceleration cost may not allow the robot to slow down as fast as needed, impacting the whole transmission performance and energy consumption.

We use a second-order motion model and a more complete motion cost that incorporates the acceleration while posing the whole setup in the continuous domain.

This continuous time reformulation enables us to pose the problem in an optimal control framework amenable to the use of maximum principle and compute the optimal control via an effective algorithm. Specifically, we use the Hamiltonian-based algorithm proposed in [43] which displayed fast and effective convergence for such optimal control problems [38]. This algorithm, summarized in the sequel, is especially suitable for continuous-time problems whose Hamiltonian has special properties. We demonstrate the efficacy of our algorithm by testing it on the considered problem and recording the CPU times required to compute the optimal control. We start with the formulation of the problem in the next section.

6.1 Problem Formulation

In this section, we first present our probabilistic channel prediction framework that allows the robot to predict the channel quality at unvisited locations along the trajectory based on a small number of a priori Channel to Noise Ratio (CNR)[1] measurements in the same environment. Then, we show how the robot can assess its average communication energy cost, given a required receiver reception quality. Next we discuss the general motion cost model that incorporates the effects of acceleration. Finally, we pose the co-optimization problem in the optimal control framework.

6.1.1 Probabilistic Channel Prediction Framework [49, 52]

In the communication literature [35], it is well-known that the CNR can be modeled as a multi-scale random process with three components: path loss, shadowing and multipath fading. Thus, we can assess the channel quality probabilistically, at an unvisited location, based on only a small number of a priori channel measurements.

In [52], Mostofi, et. al showed that a Gaussian random variable, $\gamma_{\mathrm{dB}}(q)$, can best characterize the CNR (in the dB domain) at an unvisited location q, where the mean

[1] In this work, CNR is defined as the channel power divided by the receiver noise power. Then, the received Signal to Noise Ratio will be CNR times the transmitted power.

and variance of $\gamma_{\mathrm{dB}}(q)$ are given as follows:

$$\overline{\gamma}_{\mathrm{dB}}(q) = H_q\mu + \Psi^{\mathrm{T}}(q)\Phi^{-1}\big(Y - H_{\mathcal{Q}}\mu\big), \tag{94}$$

$$\Sigma(q) = \xi^2_{\mathrm{dB}} + \rho^2_{\mathrm{dB}} - \Psi^{\mathrm{T}}(q)\Phi^{-1}\Psi(q). \tag{95}$$

Here, Y is the stacked vector of m a priori-gathered CNR measurements, $\mathcal{Q} = \big\{q_1, \cdots, q_m\big\}$ denotes the set of the measurement positions, $H_q = [1 \ -10\log_{10}(\|q - q_b\|)]$, $H_{\mathcal{Q}} = [\mathbf{1}_m \ -D_{\mathcal{Q}}]$, $\mathbf{1}_m$ represents the m-dimensional vector of all ones and q_b is the position of the remote station. Furthermore, $D_{\mathcal{Q}} = \big[10\log_{10}(\|q_1 - q_b\|) \cdots 10\log_{10}(\|q_m - q_b\|)\big]^{\mathrm{T}}$, $\Phi = \Omega + \rho^2_{\mathrm{dB}}\, I_m$ with Ω denoting a matrix with entries $\big[\Omega\big]_{i,j} = \xi^2_{\mathrm{dB}} \exp(-\|q_i - q_j\|/\beta)$, for $i, j \in \{1, \cdots, m\}$, and $\Psi(q) = \big[\xi^2_{\mathrm{dB}} \exp(-\|q - q_1\|/\beta) \ \cdots \ \xi^2_{\mathrm{dB}} \exp(-\|q - q_m\|/\beta)\big]^{\mathrm{T}}$. Finally, $\mu = [\alpha \ \ n_{\mathrm{PL}}]^{\mathrm{T}}$, ξ_{dB}, β and ρ_{dB} represent the channel parameters. Essentially, this channel prediction framework models the wireless channel as a non-stationary Gaussian random process. Then, the CNR at an unvisited location can be predicted by conditioning on the available a priori measurements in the same environment. See [49, 52] for more details on how the underlying parameters are estimated and the performance of this framework with real data and in different environments.

6.1.2 Communication Energy Model

Assuming the commonly used MQAM modulation for the communication from the robot to the remote station, the required transmit power at time t can be found as follows [35]:

$$\widetilde{P}_{\mathrm{comm}}(t) = (2^{R(t)} - 1)/(K\Upsilon(q(t))),$$

where

$$K = -1.5/\ln(5p_{b,\mathrm{th}}),$$

and $p_{b,\mathrm{th}}$ is the given required Bit Error Rate (BER) threshold (i.e. the required reception quality), $R(t)$ denotes the spectral efficiency at time t, $q(t)$ is the position of the robot at time t, and $\Upsilon(q(t))$ represents the instantaneous CNR at $q(t)$.

As discussed in Section 6.1.1, the CNR at unvisited location $q(t)$ is unknown but can be predicted as a lognormal random variable with its mean and variance given by (94) and (95) respectively. As a result, the anticipated communication power is also random. We then have the following for $P_{\text{comm}}(t)$, the average communication power (averaged over the predicted channel), which we use in the rest of the chapter:

$$P_{\text{comm}}(t) = \frac{2^{R(t)} - 1}{K} E\left[\frac{1}{\gamma(q(t))}\right]. \tag{96}$$

Note that for lognormally distributed $\gamma(q(t))$, we have

$$E\left[\frac{1}{\gamma(q(t))}\right] = \exp\left(\left(\frac{\ln 10}{10}\right)^2 \frac{\Sigma(q(t))}{2}\right) \frac{1}{\overline{\gamma}(q(t))}, \tag{97}$$

where $\overline{\gamma}(q(t)) = 10^{\overline{\gamma}_{\text{dB}}(q(t))/10}$. As can be seen, equation (97) is a measure of the predicted channel quality at $q(t)$. We say the predicted channel quality is high (low) if (97) is small (large).

6.1.3 Motion Energy Model

Suppose that the robot is moving with a velocity $v(t)$ and an acceleration $u(t)$ along the pre-defined trajectory due to an applied force generated by the DC motor of the robot. Without loss of generality, $v(t)$ and $u(t)$ are scalars, since we only need to consider the magnitude and direction of the acceleration and velocity along the trajectory. Then the motion power consumed in the process is given by [72]

$$P_{\text{mo}}(t) = k_1 u(t)^2 + k_2 v(t)^2 + k_3 v(t) + k_4 + k_5 u(t) + k_6 u(t)v(t), \tag{98}$$

where k_i, $i \in \{1, \cdots, 6\}$, is a constant. Note that (98) includes the impact of both velocity and acceleration. Also, the motion model used in [84] is a special case of (98) when k_1, k_5 and k_6 are equal to 0.

6.1.4 Optimal Control Problem

Consider a second-order dynamic model for the motion of robot as follows: $\dot{x} = v(t)$ and $\ddot{x}(t) = u(t)$. We define the state variables as $x_1 = x$ and $x_2 = \dot{x}$ and take the

acceleration $u(t)$ as the control variable. We get the following state equations:

$$\dot{x_1}(t) = x_2(t),$$
$$\dot{x_2}(t) = u(t).$$

Note that x_1 is a scalar which represents the total distance traveled along the pre-defined trajectory. The force $F(t)$ required to produce this acceleration $u(t)$ is $F(t) = Mu(t) + g(x_1(t), x_2(t))$ where $g(x_1(t), x_2(t))$ is the force incorporating the effects of air drag, friction and gravity along the path of travel, and M is the mass of the robot.

We then have the following continuous-time formulation of the co-optimization problem: Minimize

$$J_1 = \int_0^{t_f} \left(\frac{2^{R(t)} - 1}{K} s(x_1(t)) + k_1 u(t)^2 + k_2 x_2(t)^2 + k_3 x_2(t) + k_4 + k_5 u(t) + k_6 u(t) x_2(t) \right) dt \tag{99}$$

subject to dynamical constraints

$$\dot{x}_1(t) = x_2(t), \qquad x_1(0) = a, \quad x_1(t_f) = b,$$
$$\dot{x}_2(t) = u(t), \qquad x_2(0) = 0, \quad x_2(t_f) = d,$$

and constraints on the input

$$\int_0^{t_f} R(t) dt = \frac{Q}{B},$$
$$0 \leq R(t) \leq R_{\max},$$
$$-u_{\max} \leq u(t) \leq u_{\max}$$

where the control input consists of acceleration $u(t)$ and spectral efficiency $R(t)$, for $t \in [0, t_f]$, t_f is the total time budget, $s(x_1(t)) = E\left[\frac{1}{\gamma(q(x_1(t)))}\right]$ with $q(x_1(t))$ denoting the mapping from $x_1(t)$ to the corresponding position along the pre-defined trajectory, $u_{\max}$ and $R_{\max}$ denote the maximum achievable acceleration and spectral efficiency respectively, Q is the total number of information bits to be sent and B is

the given communication bandwidth. Since the robot travels along a fixed trajectory, we consider the case where the robot starts from position a and needs to move to position b at the end of operation. Moreover, we assume that the robot is stationary initially, i.e. $x_2(0) = 0$, and needs to reach velocity d at time t_f. Note that constraint $\int_0^{t_f} R(t)dt = Q/B$ guarantees that the information bits are sent to the remote station within the given time budget.

To pose the integral constraint in (99) in a way amenable to compute the optimal control, we define the auxiliary state variable x_3 in the following way:

$$\dot{x}_3 = R, \qquad x_3(0) = 0, \quad x_3(t_f) = \frac{Q}{B} := c. \tag{100}$$

The problem now becomes that of minimizing J_1 defined in (99) subject to the dynamic constraints defined via (99) and (100), as well as pointwise input constraints of the form $|u(t)| \leq u_{\max}$ and $0 \leq R(t) \leq R_{\max}$. We handle the terminal constraints in (99) and (100) by using penalty functions comprised of quadratic terms. Hence, for constants $C_1 \geq 0$, $C_2 \geq 0$, and $C_3 \geq 0$, the optimal control problem has the form: Minimize

$$\begin{aligned} J_2 = \int_0^{t_f} \left(\frac{2^R - 1}{K} s(x_1) + k_1 u^2 + k_2 x_2^2 + k_3 x_2 + k_4 + k_5 u + k_6 u x_2 \right) dt \\ + C_1 ||x_1(t_f) - b||^2 + C_2 ||x_2(t_f) - d||^2 + C_3 ||x_3(t_f) - c||^2 \end{aligned} \tag{101}$$

subject to the dynamical constraints

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \qquad x_1(0) = a, \\ \dot{x}_2(t) &= u(t), \qquad x_2(0) = 0, \\ \dot{x}_3(t) &= R(t), \qquad x_3(0) = 0, \end{aligned}$$

and constraints on the input

$$\begin{aligned} -u_{\max} \leq u(t) \leq u_{\max}, \\ 0 \leq R(t) \leq R_{\max}. \end{aligned}$$

While this simplifies the algorithm, it behooves us to ensure that the computed optimal control has its final state $x(t_f)$ be close enough to its specifications defined in (99) and (100) by appropriate selection of constants C_1, C_2 and C_3.

6.2 Computing the Optimal Controls

Consider the abstract optimal control problem whose state equation is

$$\dot{x} = f(x,u), \qquad x_0 := x(0) \ \text{ is given}, \tag{102}$$

where $x \in \mathcal{R}^n$ and $u \in \mathcal{R}^k$; the scalar-valued cost functional to be minimized is

$$J = \int_0^{t_f} L(x,u)dt; \tag{103}$$

and the input $u(t)$ has pointwise constraints of the form $u(t) \in U$ for a compact set $U \subset \mathcal{R}^k$. Suppose that the functions $f : \mathcal{R}^n \times \mathcal{R}^k \to \mathcal{R}^n$ and $L : \mathcal{R}^n \times \mathcal{R}^k \to \mathcal{R}$ are piecewise-continuously differentiable and locally Lipschitz continuous, and the final time t_f is given and fixed. Let $p(t) \in \mathcal{R}^n$ denote the costate (adjoint) equation, and recall that the Hamiltonian is defined via the following equation:

$$H(x,u,p) = p^{\mathrm{T}} f(x,u) + L(x,u), \tag{104}$$

whose Right-Hand Side's dependence on t is suppressed in the notation used.

Suppose that for every $x \in \mathcal{R}^n$ the function $f(x,u)$ is affine in u and the function $L(x,u)$ is convex in u; note that this does not mean that f is affine or L is convex jointly in both variables (x,u). Now let $u(t)$, $t \in [0,t_f]$, be an admissible control, namely a Lebesgue-measurable, absolutely integrable function from $[0,t_f]$ into U. Let $x(t)$ and $p(t)$ be the corresponding state trajectory and costate trajectory. Consider the Hamiltonian $H(x(t),u(t),p(t))$ as a function of $u(t) \in U$. Notice that $x(t)$ and $p(t)$ were derived from $u(t)$. For every $t \in [0,t_f]$, let $u^\star(t) \in \ \arg\min(H(x(t),w,p(t))|w \in U)$ which we call a (pointwise) *minimizer of the Hamiltonian*. Observe that it need not be unique. Suppose that there exists an admissible control $u^\star(t)$ such that at

every time $t \in [0, t_f]$, $u^\star(t)$ is a minimizer of the Hamiltonian in this fashion. The algorithm we next describe is suitable for problems where it is easy to compute the pointwise minimizer of the Hamiltonian.

Let $u(t)$ be a given admissible control, let $u^\star(t)$ be a minimizer of the Hamiltonian at each $t \in [0, t_f]$, and consider the control $u_{\text{next}}(t)$ defined in the following way.

1. Compute $\theta(u)$ defined as

$$\theta(u) = \int_0^{t_f} \big(H(x, u^\star, p) - H(x, u, p)\big) dt. \tag{105}$$

2. Compute the non-negative integer $k(u)$ defined as

$$k(u) := \min \big(k = 0, 1, 2, \dots \mid J(u + 2^{-k}(u^\star - u)) - J(u) \leq 2^{-(k+1)}\theta(u)\big), \tag{106}$$

   and define $\lambda(u) = 2^{-k(u)}$.

3. Set $u_{\text{next}} = u + \lambda(u)(u^\star - u)$.

The algorithm then repeats steps 1-3 until it converges.

Observe that this algorithm searches for u_{next} in the direction of $u^\star$ from u, and it is not a gradient descent but a sort of gradient projection. The stepsize yielding $u^\star$ in that direction is due to Armijo, and it is known to provide effective convergence to local minima, when used in conjunction with gradient-descent algorithms in the setting of nonlinear programming (see, e.g., [57] and references therein). In particular, these algorithms typically move quickly towards optimum points at the initial phases of their runs. The details of the Hamiltonian algorithm can be found in our work [38].

6.3 Minimizer of the Hamiltonian

In the context of our energy-aware problem defined in (101), recall that the state variable is $(x_1(t), x_2(t), x_3(t))$ and the input is $(u(t), R(t))$, and let $p(t) = (p_1(t), p_2(t), p_3(t))$

be an explicit coordinate representation of the costate. By the costate equation, they have the forms

$$\begin{aligned}
\dot{p}_1 &= -\frac{2^R-1}{K}\frac{\partial s(x_1)}{\partial x_1}, & p_1(t_f) &= 2C_1(x_1(t_f)-b),\\
\dot{p}_2 &= -p_1 - 2k_2x_2 - k_3 - k_6u, & p_2(t_f) &= 2C_2(x_2(t_f)-d),\\
\dot{p}_3 &= 0, & p_3(t_f) &= 2C_3(x_3(t_f)-c),
\end{aligned} \tag{107}$$

and by (104), the Hamiltonian is given by

$$\begin{aligned}
H(x,u,p) = p_1x_2 + p_2u + p_3R + \frac{2^R-1}{K}s(x_1) + k_1u^2\\
+ k_2x_2^2 + k_3x_2 + k_4 + k_5u + k_6ux_2,
\end{aligned} \tag{108}$$

where the dependence on t is suppressed in the notation. It is readily seen that the minimizer of the Hamiltonian is given by

$$u^\star = \begin{cases}
-\dfrac{(p_2+k_5+k_6x_2)}{2k_1} & \text{if } \left|\dfrac{(p_2+k_5+k_6x_2)}{2k_1}\right| \le u_{\max},\\
u_{\max} & \text{if } -\dfrac{(p_2+k_5+k_6x_2)}{2k_1} > u_{\max},\\
-u_{\max} & \text{if } -\dfrac{(p_2+k_5+k_6x_2)}{2k_1} < -u_{\max},
\end{cases}$$

and

$$R^\star = \begin{cases}
\dfrac{1}{\ln(2)}\ln\left(\dfrac{-p_3K}{\ln(2)s(x_1)}\right) & \text{if } p_3 \le -\dfrac{(\ln(2)s(x_1))}{K},\\
R_{\max} & \text{if } \dfrac{1}{\ln(2)}\ln\left(\dfrac{-p_3K}{\ln(2)s(x_1)}\right) > R_{\max},\\
0 & \text{otherwise.}
\end{cases}$$

We observe that $R^\star$ is lower at points where $s(x_1)$ is higher and vice versa. This means that the robot should transmit at higher (lower) rates in places where the channel quality is higher (lower), as expected. All but the terms $s(x_1) = E\left[\frac{1}{\gamma(q(x_1))}\right]$ and its derivative $\frac{\partial s(x_1)}{\partial x_1}$ in the RHS of (127) and (108) are explicit and can be easily calculated. $s(x_1)$ and its derivate can then be calculated by utilizing (97).

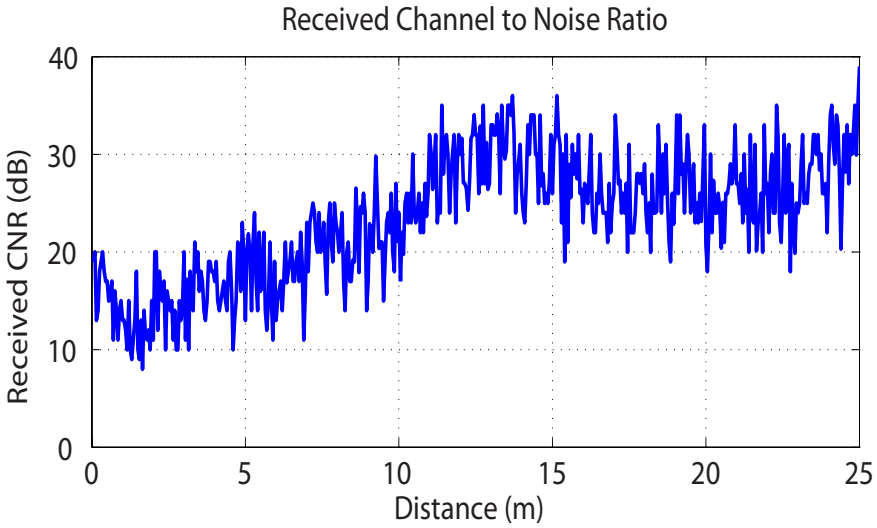


Figure 39: The figure shows the CNR (based on real measurements) along a straight line with the length of 25 meters. The channel quality is predicted by taking few samples from these measurements.

6.4 Simulation Results

We coded our algorithm in MATLAB and ran it on a laptop with Intel dual core i5 3317U 1.7 GHz processor and 4GB 1600 MHz RAM. The problem we considered has the following parameters. The robot travels in a straight line with length of 25 meters in 20 seconds, and hence $t_f = 20$, $x_1(0) = 0$ and $x_1(t_f) = 25$. It moves towards the transmitter located 40 m from its initial placement, i.e. $q_b = 40$. The robot experiences the CNR of Fig. 39 across its path, which is based on real channel measurements (see [49] for details on the measurement setup). The channel is mainly unknown to the robot and is predicted based on 20% a priori measurements in the same environment by using the channel prediction framework of [49, 52], summarized in Section 6.1.1. This results in the following channel parameters: $\alpha = -41.34$, $n_{\text{PL}} = 3.86$, $\xi_{\text{dB}} = 3.20$, $\beta = 3.09$ meters, and $\rho_{\text{dB}} = 2.77$. We used these to compute offline $s(x_1(t))$ and its derivative. We set the total number of information bits to be $Q/B = 100$ bits/Hz (i.e. $x_3(t_f) = 100$), and $R_{\max} = 6$ bits/Hz/s.

The motion cost parameters in (98) are set to $k_1 = 5.47$, $k_2 = 0.77$, $k_3 = 10.10$,

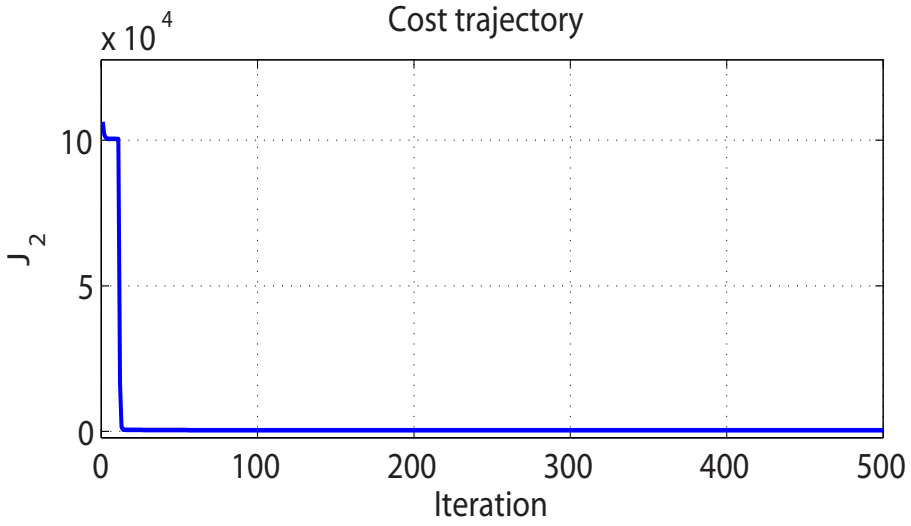


Figure 40: The plot shows the anticipated total energy cost (J_2) as a function of iteration step. The cost almost reduces to its final value in about 20 iterations.

$k_4 = 4.24$, $k_5 = 0$ and $k_6 = 0$ as in [72][2]. The upper bound on the acceleration is taken as $u_{\max} = 1\,\mathrm{m/s^2}$. Moreover, the initial conditions in (101) are chosen as $x_2(0) = x_3(0) = 0$, the terminal velocity is chosen as $x_2(t_f) = 0$, and the penalty weights of J_2 are $C_1 = 10$, $C_2 = 10$ and $C_3 = 10$.

The algorithm was run for 500 iterations, and the results are shown in Figures 40-44. All differential equations were solved via the forward Euler method based on the time-difference $\Delta t = 0.001$s. Fig. 40 plots the graph of the cost J_2 as a function of iteration count. It shows that much of the cost reduction occurs in about 20 iterations. Such rapid descent in the early stages of an algorithm's run are characteristic in gradient-descent algorithms with Armijo step sizes [57], and is one of the reasons we chose it as part of our computational technique. As a matter of fact, the initial cost is $J_2 = 1.0633 \times 10^5$ and after 20 iterations the cost is $J_2 = 540.54$. The cost at the end of 500 iterations is $J_2 = 402.47$. To gain a clearer view of the cost-descent, we plot the graph of J_2 from iteration 20 onwards in Fig. 41.

[2]Note that we can consider $k_5 = 0$ and $k_6 = 0$ if the initial and final velocities of the robot are the same [72].

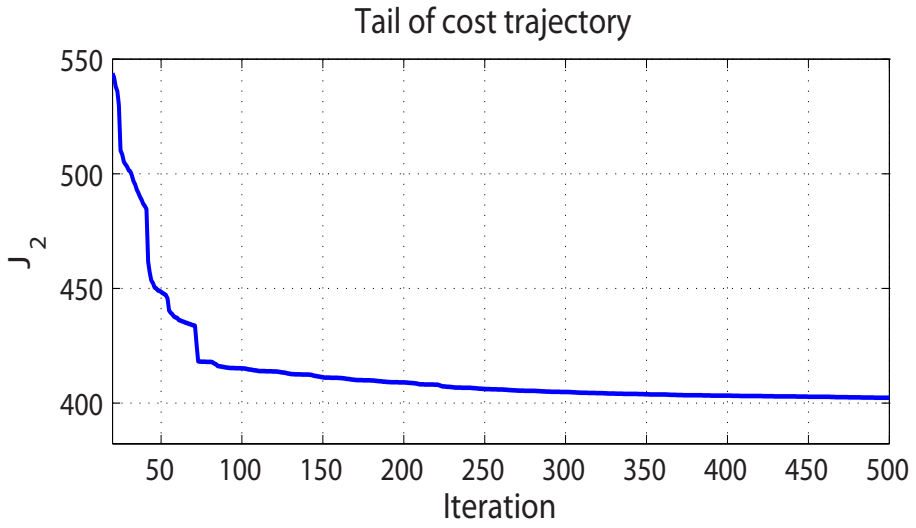


Figure 41: The plot shows the tail of the curve in Fig. 40. The cost remains almost constant after the first few iterations.

The term $\theta(u)$ acts as an *optimality function*, namely a measure of the extent to which u fails to be an optimal control (see [57] for an extensive discussion on optimality functions and their role in optimization). To see this, observe that $\theta(u)$ is always non-positive since $u^\star$ minimizes the Hamiltonian. The condition $\theta(u) = 0$ is equivalent to the maximum principle, and $|\theta(u)|$ can be viewed as a measure of u failing to satisfy the maximum principle. Generally, a characterization of an algorithm's convergence can be based on how close $|\theta(u)|$ gets to zero (see [57]). As for our algorithm, Fig. 42, showing $\theta(u)$ as a function of iteration counts, indicates together with the tail of the graph in Fig. 41 that the algorithm has converged. We also mention that the final-state constraints of $x_1(t_f) = 25$, $x_2(t_f) = 0$ and $x_3(t_f) = 100$ are almost met, with the obtained values of $x_1(t_f) = 24.42$, $x_2(t_f) = 0.21$ and $x_3(t_f) = 99.91$. Choosing larger values of C_1, C_2 and C_3 only makes the asymptotic convergence slower without any significant gains on the accuracy of final states, which are already very close to the desired values for the given choices of C_1, C_2 and C_3.

Remark. Due to the disparate nature of our control variables u and R we use block iterations in the algorithm, namely, we cycle between 10 iterations in u for a

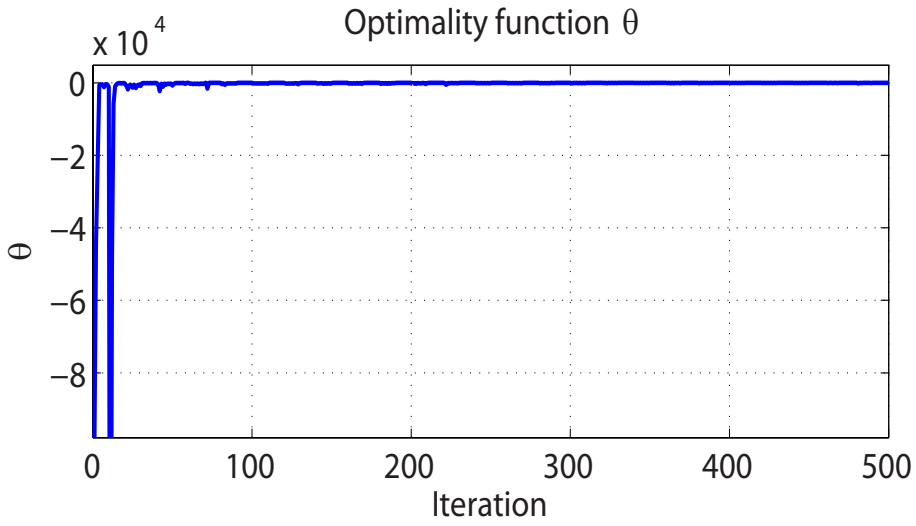


Figure 42: Optimality function quickly goes to zero as the algorithm converges. The magnitude of optimality function is a measure of how close we are to the optimal solution.

fixed R and then 10 iterations in R for a fixed u. The effect of this can be seen for example in Fig. 3, where the cost-reduction is more pronounced at the start of a block than at its end.

Fig. 43 depicts the computed optimal controls and robot behaviors at different points along the path while Fig. 44 shows their evolution in time domain. It can be seen that periods of higher predicted channel quality (low ($s(x_1)$)) correspond to larger spectral efficiency (R) as one would have expected. Moreover, the spectral efficiency changes instantaneously to changes in channel quality. The velocity response is a bit delayed to changes in channel quality and is quite smooth. This can be ascribed to using acceleration as an input and defining a cost on velocity. Again, the acceleration profile is very smooth due a cost defined on it. This behavior stands in contrast to the findings of [84] which, using the velocity as a control in a (discrete-time) co-optimization problem, yields wider variations in velocity $v(t)$, and hence large accelerations $|u(t)|$. In practical situations, the acceleration may be limited due to operational constraints and hence its explicit inclusion in the optimal control problem appears to be a more practical approach. To bring the final velocity to zero, we see

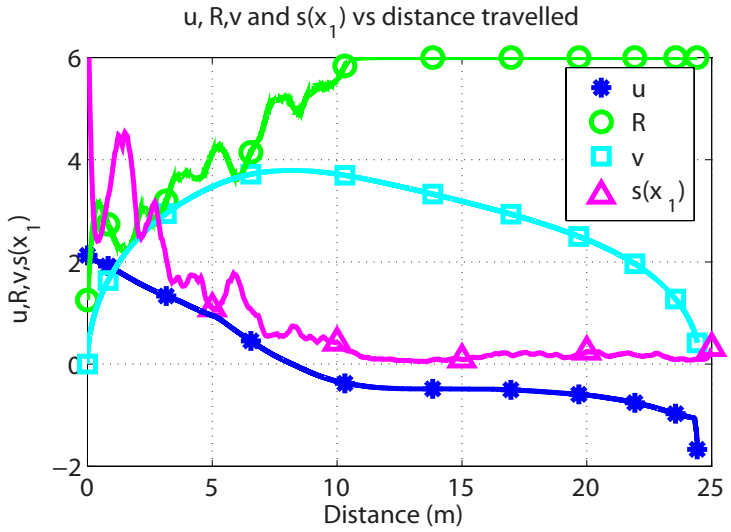


Figure 43: The acceleration u, spectral efficiency R, velocity $v = x_2$ and channel $s(x_1)$ along the path of travel. u and v are magnified by factor 5 and 2 respectively while $x_1(t)$ and $s(x_1)$ are scaled by factor of 8 and 2 respectively. We can clearly see how the robot adapts its communication and motion strategies based on the predicted channel quality metric ($s(x_1)$). The delay caused by the acceleration cost can also been observed.

a sharp deceleration towards the end.

To test the efficiency of our approach, we ran the algorithm for different time steps used in the numerical integration, as well as different numbers of iterations. The results for $\Delta t = 0.001, 0.005, 0.01$ seconds, and for 250 and 500 iterations, are shown in Table 3.

While we observe significant changes in the CPU times of the three runs, we note that the final results of the computations are quite close to each other. In particular for $\Delta t = 0.01$ s and 250 iterations, the algorithm's run was completed in 3.49 seconds on CPU time. This indicates the potential use of our algorithm for real time applications and this is the topic of our next chapter.

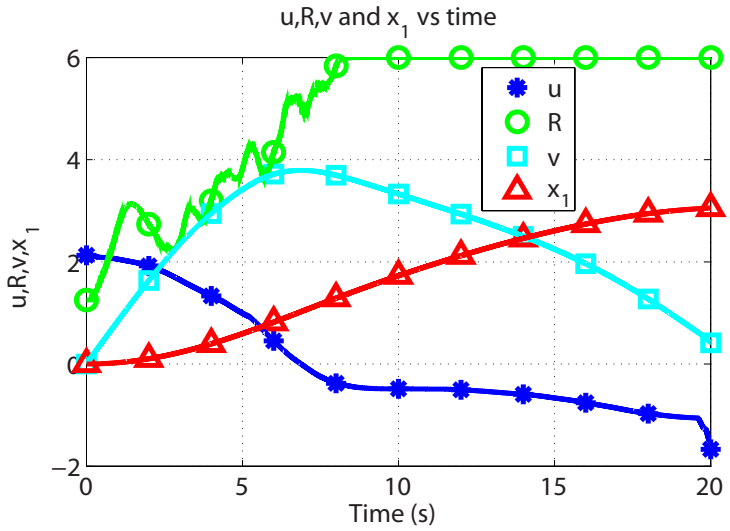


Figure 44: The acceleration u, spectral efficiency R, velocity $v = x_2$ and distance traveled x_1 vs time. u and v are magnified by factor 5 and 2 respectively while $x_1(t)$ is scaled by factor 8. We can clearly see how the robot adapts its communication and motion strategies based on the predicted channel quality metric ($s(x_1)$). The delay caused by the acceleration cost can also been observed.

Table 3: Results of simulations for various steps sizes and number of iterations. Using large step size in integration reduces the compute time without having much impact on the final cost or terminal state constraints, indicating the potential for real-time like application which we consider in the next chapter.

Δt	iterations	Final Cost	CPU Time	Final Position	Final Velocity	Bits/Hertz
0.001	500	402.40	56.23	24.42	0.21	99.91
0.001	250	406.20	28.03	24.43	0.40	99.90
0.005	500	402.50	12.35	24.44	0.20	99.89
0.005	250	405.50	6.18	24.44	0.33	99.88
0.010	500	403.50	7.10	24.42	0.25	99.90
0.010	250	406.70	3.49	24.43	0.40	99.86
Initial Cost = 1.0633e5; CPU time in seconds						

CHAPTER VII

REAL-TIME MOTION AND COMMUNICATION CO-OPTIMIZATION IN PLANE

This chapter extends the problem and methodology developed in the previous chapter in the following two ways. First, it adds the challenging element of path planning by lifting the restriction that the robot has to follow a pre-determined path. Rather, it has to compute an optimal trajectory. Second, channel prediction is updated as the robot gathers more channel samples in the workspace, requiring a re-evaluation of the optimal trajectory for the cost to go, in contrast to the problem in previous chapter which does a one-time channel prediction and co-optimization before the robot starts moving. We point out that we do not use model-predictive control or rolling horizons, but rather compute the entire trajectory of the cost-to-go performance functional to the given final time. These enhancements over the previous setting pose significant computational challenges and the proposed scheme handles these challenges in an effective way, as will be demonstrated by its application to example problems.

Similarly to the work in previous chapter, the objective is to minimize a weighted sum of communication and motion energy costs given the number of bits to be transmitted. The starting and final times, as well as the initial and final locations of the robot are given. Different from the problem considered in chapter 6 however is that the robot's trajectory is optimized over the entire 2D workspace, which is closely-coupled with the spatial variation of communication channel quality. Theses additional elements in the optimization greatly complicate the problem. Secondly, this work makes a first attempt to solve the problem in a real-time setting.

The spatial channel quality over the 2D workspace is not known beforehand but

rather updated during the operation as more channel measurements become available to the robot. Utilizing the probabilistic framework of [49,52], the channel parameters can be estimated and the channel quality over the space predicted. With updated channel information, the optimization process is also updated correspondingly. This defines an online framework for the optimization process, which is made possible by the fast convergence of the optimization algorithm and its ability to react rapidly to changes in the estimated channel fading. The obtained simulation results hold out promise of its eventual implementation in future realistic scenarios. We start with the formulation of the problem in the next section.

7.1 Problem Definition.

Consider a robot that has to traverse a path between a source point $S \in \mathcal{R}^2$ and a destination point $D \in \mathcal{R}^2$ while transmitting a given number of bits to a remote station in a given time-horizon $[0, t_f]$. The problem is to determine the robot's path, acceleration, and transmission rate as functions of time $t \in [0, t_f]$ so as to minimize the total energy required for transmission and motion. Based on few measurements, the channel quality is predicted probabilistically as will be detailed below. The power required for motion depends on the robot's velocity and acceleration, while its transmission power depends on its position relative to the remote station, transmission rate, and the channel quality (shadowing and multipath fading). For the robot's motion, we use the second order dynamical model

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= u(t), \end{aligned} \tag{109}$$

where $x_1 \in \mathcal{R}^2$ is the position of the robot in the plane, $x_2 \in \mathcal{R}^2$ denotes its velocity, and $u \in \mathcal{R}^2$ is its acceleration. The initial condition of this equation is $x_1(0) = S$ and $x_2(0) = 0$. According to Ref. [72], the power required for the robot's motion has the

form, for given constants $k_i \geq 0$, $i = 1, \ldots, 6$,

$$P_m(t) = k_1||u(t)||^2 + k_2||x_2(t)||^2 + k_3||x_2(t)|| + k_4 + k_5||u(t)|| + k_6||u(t)|| \cdot ||x_2(t)||. \tag{110}$$

The power required for transmitting data to the remote station (from position $x_1(t)$) is given by

$$P_c(t) = \frac{2^{R(t)} - 1}{K} s(x_1(t)), \tag{111}$$

where $R(t) \geq 0$ is the spectral efficiency of the channel at time t and position $x_1(t)$, K is a constant depending on the threshold bit error rate acceptable at the receiver, and $s(x_1(t))$ is the estimated channel quality metric at position $x_1(t) \in \mathcal{R}^2$; see the section on channel estimation in previous chapter. Let Q be the total number of bits the robot has to transmit, then the requirement of transmitting Q bits in time interval $[0, t_f]$ results in the constraint

$$\int_0^{t_f} R(t)dt = \frac{Q}{B} := c,$$

where B denotes the channel's bandwidth. To get rid of the integral, so as to make this constraint more amenable to our algorithm, we introduce an auxiliary state variable, $x_3 \in \mathcal{R}$, defined by the equation

$$\dot{x}_3 = R(t), \tag{112}$$

with the boundary conditions $x_3(0) = 0$ and $x_3(t_f) = c$. Other final-time constraints on the state variable (position and velocity) are $x_1(t_f) = D$ and $x_2(t_f) = 0$. We also assume upper-bound constraints on $u(t)$ and $R(t)$ of the form

$$0 \leq ||u(t)|| \leq u_{\max}, \quad 0 \leq R(t) \leq R_{\max}, \tag{113}$$

for given $u_{\max} > 0$ and $R_{\max} > 0$.

The related optimal control problem is defined as follows. Its input is $(u(t), R(t)) \in \mathcal{R}^2 \times \mathcal{R}$, $t \in [0, t_f]$, its state is $(x_1(t), x_2(t), x_3(t))$, and its dynamics are given by Eqs.

(109) and (112) with the initial conditions $x_1(0) = S$, $x_2(0) = 0$, and $x_3(0) = 0$. Its performance function, to be minimized, is

$$\bar{J} := \int_0^{t_f} \big(P_m(t) + \gamma P_c(t)\big)dt, \tag{114}$$

where $P_m(t)$ and $P_c(t)$ are the motion power and transmission power defined, respectively, by Eqs. (110) and (111), and $\gamma > 0$ is a given constant. The problem is to minimize $\bar{J}$ subject to the above dynamic equations, the upper-bound constraints on the input as defined by Eq. (113), and the final-state constraints $x_1(t_f) = D$, $x_2(t_f) = 0$, and $x_3(t_f) = c$.

We handle the final-state constraints with a penalty function of the form $C_1||x_1(t_f) - D||^2 + C_2||x_2(t_f)||^2 + C_3||x_3 - c||^2$, for constants $C_1 > 0$, $C_2 > 0$, and $C_3 > 0$, chosen large enough so that the terminal constraints are almost satisfied. The resulting optimal control problem now has the following form: Minimize the cost functional J defined as

$$\begin{aligned} J = \int_0^{t_f} \Bigg(& \frac{2^{R(t)} - 1}{K} s(x_1) + \gamma\big(k_1||u(t)||^2 + k_2||x_2(t)||^2 \\ & + k_3||x_2(t)|| + k_4 + k_5||u(t)|| + k_6||u(t)||\dot{|}|x_2(t)||\big)\Bigg)dt \\ & + C_1||x_1(t_f) - D||^2 + C_2||x_2(t_f)||^2 + C_3||x_3(t_f) - c||^2, \end{aligned} \tag{115}$$

subject to the dynamic equations

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \qquad x_1(0) = S \\ \dot{x}_2(t) &= u(t), \qquad x_2(0) = 0 \\ \dot{x}_3(t) &= R(t), \qquad x_3(0) = 0, \end{aligned}$$

and constraints on the control inputs given by

$$\begin{aligned} &||u(t)|| \le u_{\max}, \\ &0 \le R(t) \le R_{\max}. \end{aligned}$$

We use the same channel prediction scheme and Hamiltonian based algorithm presented in the previous chapter. Based on few channel measurements collected a priori (e.g. by static sensors in the field), an initial prediction of channel quality over the workspace can be performed and the optimal control problem (140) solved for optimal controls. As the robot moves, it gathers additional channel measurements (e.g. by gathering more samples along its path, through crowdsourcing and/or by other robots in the field), which enables it to predict the channel quality more accurately. Thus the channel prediction and subsequent optimization is performed from time to time over the remaining time horizon $[t_0, t_f]$, where t_0 denotes the present time at which the optimization is performed. The details of this online optimization procedure will be presented in Section 7.4.

7.2 Path Planning with Motion and Communication Co-optimization

In this section, we consider the application of the algorithm to the problem defined in section II. The Hamiltonian associated with the optimal control problem (140) is

$$H(x,[u,R],p) = p_1^T x_2 + p_2^T u + p_3 R + \frac{2^R-1}{K} s(x_1) + \gamma\big(k_1||u||^2 + k_2||x_2||^2 + k_3||x_2|| + k_4 + k_5||u|| + k_6||u||||x_2||\big), \quad (116)$$

where the costates $p_1 \in \mathcal{R}^2$, $p_2 \in \mathcal{R}^2$, and $p_3 \in \mathcal{R}$ are defined by the adjoint equations

$$\begin{aligned}
\dot{p}_1 &= -\frac{2^R-1}{K}\frac{\partial s(x_1)}{\partial x_1}, \\
\dot{p}_2 &= -p_1 - \gamma\left(2k_2x_2 + k_3\frac{x_2}{||x_2||} + k_6||u||\frac{x_2}{||x_2||}\right), \\
\dot{p}_3 &= 0,
\end{aligned}$$

with terminal constraints $p_1(t_f) = 2C_1(x_1(t_f) - D)$, $\quad p_2(t_f) = 2C_2x_2(t_f)$ and $p_3(t_f) = 2C_3(x_3(t_f) - c)$, respectively. In the forthcoming we assume that $k_5 = k_6 = 0$ in (116) as we did in the previous chapter. The minimizer of this Hamiltonian subject to the

input constraints can be seen to be given by

$$u^{\star} = \begin{cases} -\dfrac{p_2}{2\gamma k_1}, & \text{if } \dfrac{1}{2\gamma k_1}||p_2|| \leq u_{\max} \\ -\dfrac{p_2}{||p_2||}u_{max}, & \text{if } \dfrac{1}{2\gamma k_1}||p_2|| > u_{\max}, \end{cases}$$

$$R^{\star} = \begin{cases} \dfrac{1}{\ln(2)} \ln\left(\dfrac{-p_3 K}{\ln(2)s(x_1)}\right), \text{ if } p_3 \leq -\dfrac{(\ln(2)s(x_1))}{K} \\ R_{\max}, \qquad\qquad \text{if } \dfrac{1}{\ln(2)} \ln\left(\dfrac{-p_3 K}{\ln(2)s(x_1)}\right) > R_{\max} \\ 0, \qquad\qquad \text{otherwise.} \end{cases}$$

We next consider an application example to demonstrate the effectiveness of our method for handling the real-time co-optimization problem.

7.3 Application

Consider a robot that is tasked to move from the initial point $S = (20, 40)$ to the final point $D = (10, 5)$ in the plane, and it has to transmit 150 bits/Hz to a remote station located at $q_b = (5, 5)$ in 40 seconds. The acceleration and spectral efficiency can take maximum values of $u_{max} = 0.5m/s^2$ and $R_{max} = 6\, Bits/Hz/sec$, respectively. The balancing factor between motion and communication is $\gamma = 0.01$, and the constants C_1, C_2 and C_3 are set to 10, 50 and 10, respectively. The Armijo step size parameters are set to $\alpha = 0.1$ and $\beta = 0.5$. The initial controls $u_0(t)$ and $R_0(t)$ are both set to zero. The integration step size for the simulation is set to $dt = 0.1$ seconds, and the algorithm is run for 500 iterations. The algorithm is terminated whenever the Armijo parameter k is greater than 50, indicating that a local minimum has been reached.

This robotic operation is performed under a simulated wireless channel with realistic parameters over an area of $50m \times 50m$. The channel parameters based on [49] and [6], are $K_{PL} = -41.34$, $n_{PL} = 3.86$, $\xi_{dB} = 3.20$, $\eta = 3.09m$ and $\rho_{dB} = 1.64$. The receiver thermal noise is -110 dBm and the BER threshold is set to $p_{b,th} = 2 \times 10^{-6}$. This channel can be predicted with few measurements over the field by using the

methodology summarized in previous chapter. To illustrate this point, Fig. 45 shows a sample simulated wireless channel generated with the parameters listed above, for the $250,000$ points in the plane. It is then predicted at all these points based on only 500 a priori known randomly-spaced channel samples (0.2%) over the field and the result is shown in Fig. 46. The two results are quite similar.

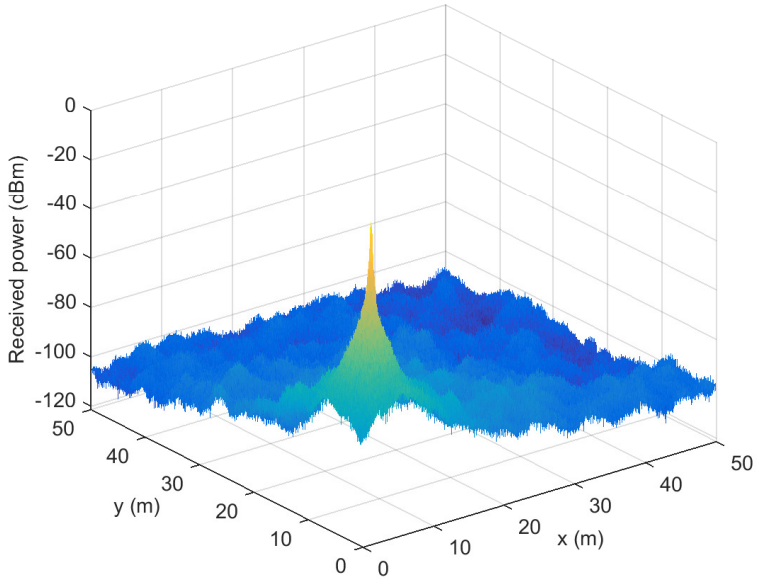


Figure 45: Simulated wireless channel over the workspace. The peak corresponds to the location of base station with which the robot is communicating.

The plot for cost (J) versus iteration count is depicted in Fig. 47, showing a rapid decrease in cost during the first few iterations of the algorithm. The cost decreases from the initial value of 2.3872×10^5 to 799.63 in 20 iterations, while the cost after 56 iterations is 565.13 when k became greater than 50, indicating convergence. Fig. 47 also shows the tail of the cost trajectory and evidently it starts flattening after iteration 20. The 56 steps of the algorithm took 0.83 seconds of CPU time on an Intel dual-core computer with i5 processor running at 2.7 GHz.

The total motion and communication cost (114), excluding the penalty term, is $\bar{J} = 475.10$, and the final values of state variables are $x_1(t_f) = (9.8, 5)$, $x_2(t_f) =$

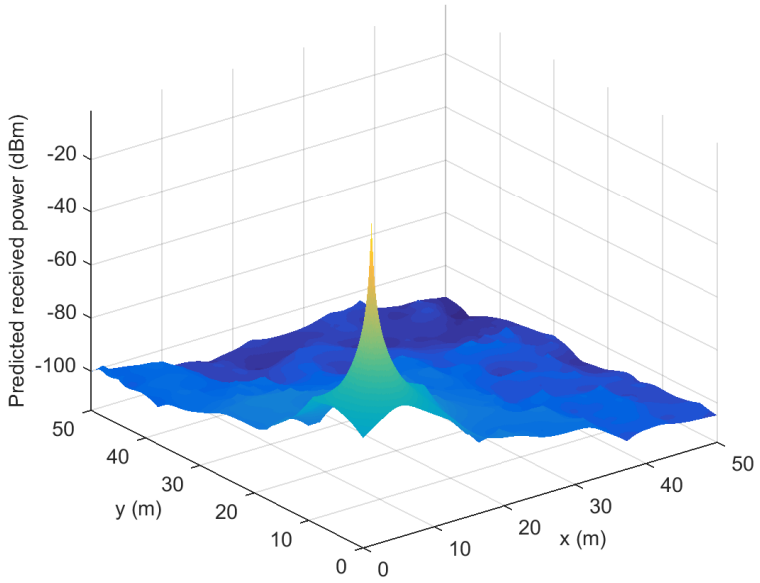


Figure 46: Predicted channel based on 500 measurements. As the number of samples for channel prediction increase, the variations away from the base station will become more apparent and the predicted channel would resemble the wireless channel of Fig. 45 more closely.

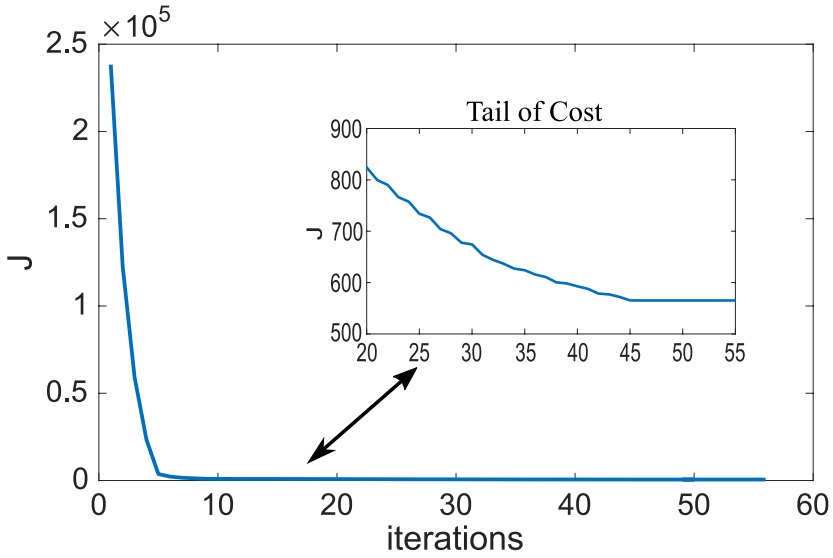


Figure 47: Cost as function of iteration count. Much of the cost decrease happens during the first few iterations. As can be seen from the tail of the cost, after iteration 20, there is not much decrease in the cost.

$(0.2, -0.8)$, and $x_3(t_f) = 149.7$. We note a mild discrepancy from the desired final values of $x_1(t_f) = (10, 5)$, $x_2(t_f) = (0, 0)$, and $x_3(t_f) = 150$, It can be reduced by choosing larger penalty terms C_1, C_2, and C_3, however choosing very large penalty terms can degrade the convergence rate. For example, setting $C_1 = 500$, $C_2 = 500$, and $C_3 = 500$ gives $x_1(t_f) = (9.99, 5)$, $x_2(t_f) = (-0.08, -0.68)$, and $x_3(t_f) = 149.99$ while the CPU time of the run increased to 7.14 seconds. It is not surprising that the initial cost is higher since the penalty terms are larger, and for the same reason, the algorithm drives the control parameters to a more restricted set and hence the final energy cost is expected to be higher as well. The CPU times often are larger in penalty-function methods with larger penalty terms.

Fig. 48 shows the log plot of predicted channel quality metric ($s(x_1) = E[1/\Upsilon(x_1)]$, where $\Upsilon(x_1)$ is the predicted received CNR at position $x_1 = (x, y) \in \mathcal{R}^2$) and the path taken by the robot in the plane. Smaller values of $s(x_1)$ correspond to good channel quality and vice versa. The robot starting and end positions are marked by a diamond and a square, respectively, in all the figures. Instead of following a straight line between them, the robot takes a detour towards areas with relatively good predicted channel quality. For instance, the point of best channel quality is $q_b = (5, 5)$, namely the location of base station, and hence the robot veers towards this point before turning away towards its destination point.

Fig. 49 depicts a three-dimensional graph of the robot's motion, where the z axis representing time and the motion is in the $x - y$ plane. The upper, blue curve represents the flow of time from 0 to 40 seconds, and the position of the robot at time t is seen by projecting the corresponding point on the upper curve onto the $x - y$ plane, where it is indicated by a corresponding point on the red curve. Fig. 50 shows the acceleration of the robot along its path, where lengths of the arrows represent its magnitude, and Fig. 51 shows the speed of the robot along its path.

The spectral efficiency of the robot along its path is shown in Fig. 52, where the

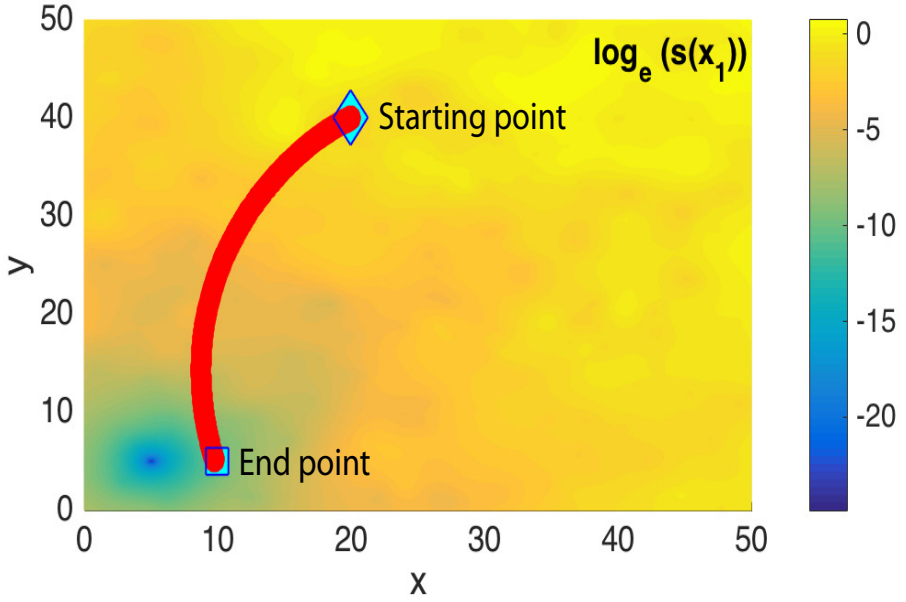


Figure 48: Path followed by the robot, veering towards regions of better channel quality. Smaller values in the colormap indicate better channel quality prediction as measured by $s(x_1)$.

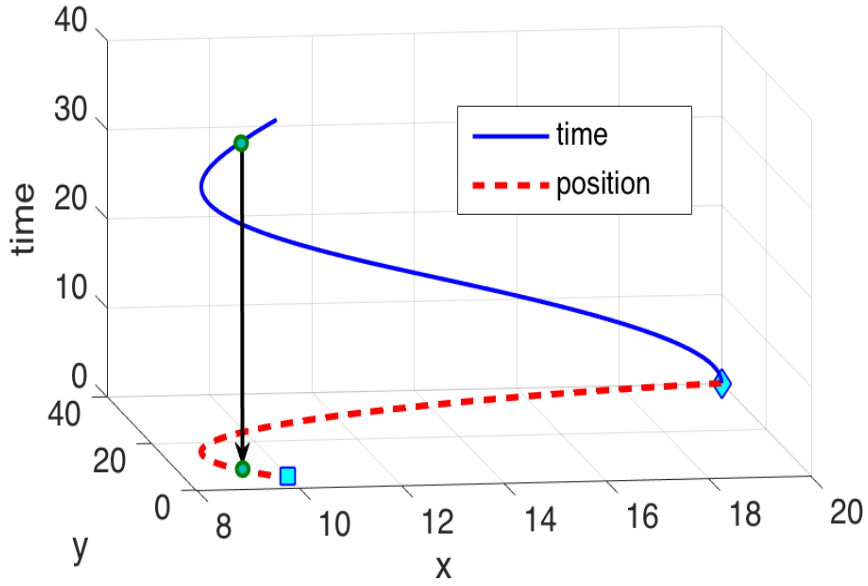


Figure 49: Position of the robot as a function of time. The diamond and the square indicate the initial and final positions respectively. Projecting an arrow from the blue curve onto the red curve gives the position of robot in the plane at the corresponding point in time

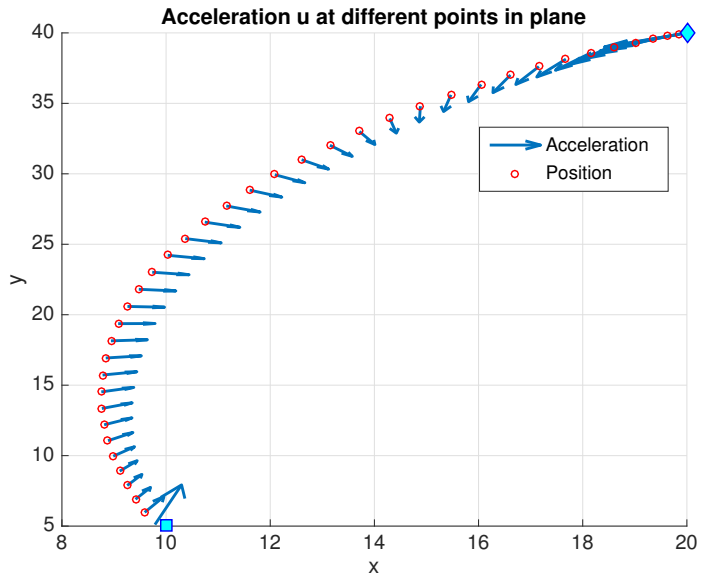


Figure 50: Acceleration of the robot along its path. The diamond and the square indicate the initial and final positions respectively. The length of the arrow shows the magnitude while the direction shows the direction of acceleration at different points along the path.

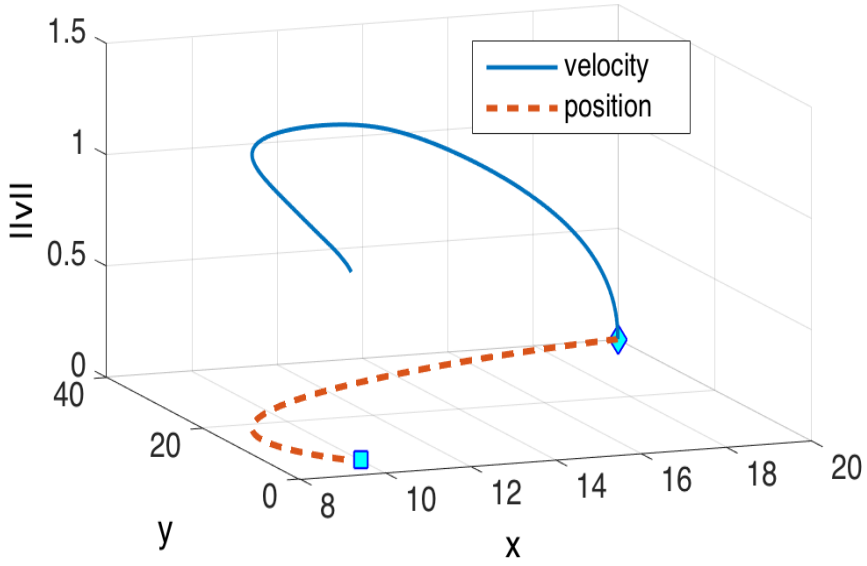


Figure 51: Velocity of the robot along its path. The diamond and the square indicate the initial and final positions respectively. Projecting an arrow from the red curve in the plane onto the blue curve along the z-axis gives the magnitude of velocity at the respective point.

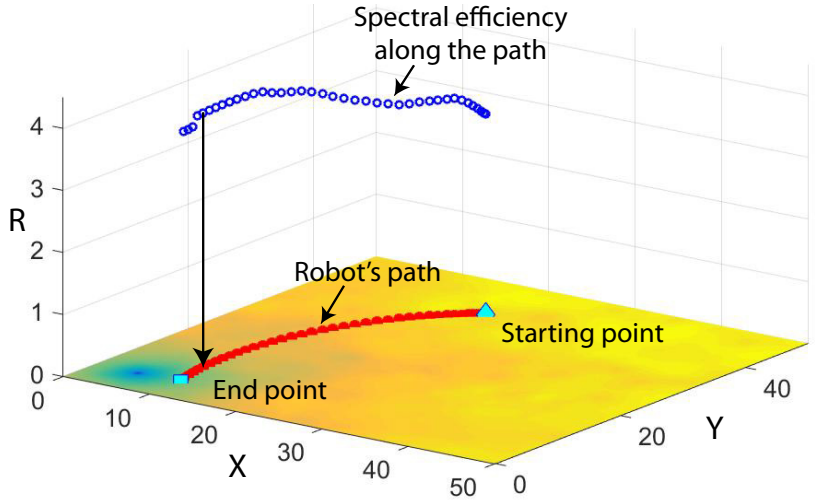


Figure 52: The robot's spectral efficiency along its path. Near the region close to the base station, the channel quality is good and the robot transmits with relatively higher spectral efficiency. At any point along the optimal path, there is an inverse relationship between the spectral efficiency and $s(x_1)$,

path is marked by red circles, and the spectral efficiency at corresponding points is marked in blue. The 2D map of the predicted channel quality metric $s(x_1)$ is also plotted on the x-y plane (in log scale). It can be seen that the robot transmits with a higher spectral efficiency in regions of better predicted channel quality. This is not surprising since, in regions of higher channel quality, the robot can transmit with a higher rate to the base station with less communication power.

7.4 Online Optimization

This section extends the algorithm to a realistic and practical online setting, where the robot obtains new channel measurements while in motion. It does not discard the older measurements, but rather appends them by the new data in order to enhance its channel prediction. The online optimal control problem is then to minimize the

cost functional

$$J_{t_0} = \int_{t_0}^{t_f} \Big(\frac{2^{R(t)} - 1}{K} s(x_1) + \gamma\big(k_1||u(t)||^2 + k_2||x_2(t)||^2 + k_3||x_2(t)|| + k_4 + k_5||u(t)|| + k_6||u(t)||||x_2(t)||\big)\Big) dt + C_1||x_1(t_f) - D||^2 + C_2||x_2(t_f)||^2 + C_3||x_3(t_f) - \bar{c})||^2, \tag{117}$$

subject to the dynamics

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \qquad & x_1(t_0) &= a_1 \\ \dot{x}_2(t) &= u(t), \qquad & x_2(t_0) &= a_2 \\ \dot{x}_3(t) &= R(t), \qquad & x_3(t_0) &= 0 \end{aligned}$$

and the input constraints and the constraints

$$\begin{aligned} 0 \leq ||u(t)|| \leq u_{\max}, \\ 0 \leq R(t) \leq R_{\max}, \end{aligned}$$

Here $t_0 \in [0, t_f]$ is the time at which the optimization is performed, and the terms a_1 and a_2 are the current position and velocity of the robot at time t_0, and $\bar{c} := (c - x_3(t_0^-))$ is the number of bits per unit frequency that remains to be transmitted in the time-interval $[t_0, t_f]$. The online approach solves this problem each time a channel estimation is performed, typically at a finite number of times during the horizon $[0, t_f]$. The initial control point of each such a run of the algorithm consists of the remaining input control computed by its previous run.

The considered problem is the same as the one discussed in Section 7.2, except that the robot performs channel prediction every 10 seconds, and each prediction is based on 100 new channel measurements taken at random locations. Also the initial run, at $t_0 = 0$, solves the offline problem with 100 channel samples. The combined time for channel prediction and a run of the algorithm was about 2 seconds and took under 50 iterations of the algorithm's run.

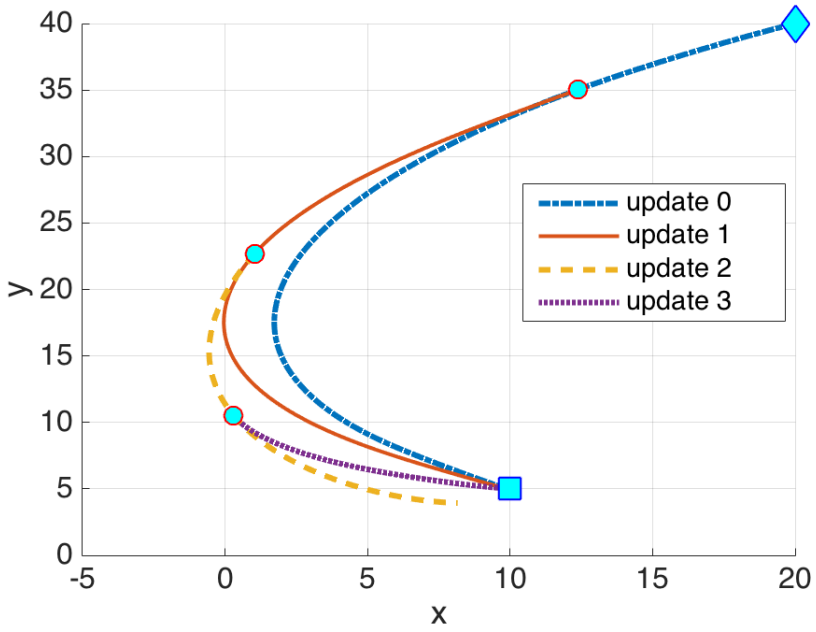


Figure 53: Online optimization after every 10 seconds. The diamond and square shows the starting and stopping position of the robot. The robot predicts the channel based on initial samples and executes its policy for the first 10 seconds till it reaches the first circle where the process is repeated.

The results of the simulation are shown in Fig. 53 & 54, where the position of the robot at the end of each predication and optimization cycle (10 seconds) is indicated by a circle. Fig. 53 shows the computed optimal trajectories for each prediction-optimization cycle from the current time to the final time. A concatenation of the computed trajectories, which the robot actually would traverse, is indicated by the red path in Fig. 54, while the dashed blue path indicates the trajectory computed by the offline algorithm at time $t_0 = 0$, based on the initial channel prediction. The total energy consumed (Eq. (114)) in the offline solution (dashed blue path Fig. 54) is $\bar{J} = 371$, while the solution of the online problem (red path in Fig. 54) yields a lower value, $\bar{J} = 304$.

7.5 MultiAgent Co-optimization in Plane

The methodology provided for the single agent case can be extended to multi-agent case to perform co-ordinated tasks and an example is provided in Fig. 55- 56. The

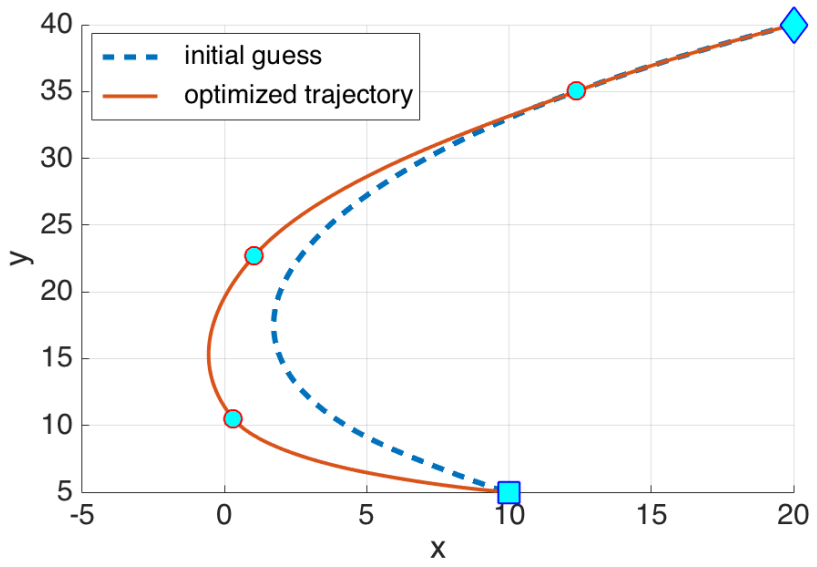


Figure 54: Offline Vs. Online optimized trajectories. The online trajectory is obtained from Fig. 53 by connecting the optimal paths between the circles in that figure.

six agents in the plane start from the same point and they have data that needs to be communicated to the base station located at $(5, 5)$ as before and in the same channel environment. Each agent, has to go to a different destination point and thus experience different channel quality and thus it would be advantageous for some agents to carry more data for transmission to base station than others. Based on some information about the channel provided by crowd-sourcing or cloud, they predict the channel and solve the problem in a centralized fashion so as to minimize the sum of all costs. The final states $x_3(t_f)$ of all the robots needs to sum up to the total data requirement and this is incorporated via a penalty term into the joint cost. The problem is solved which gives the initial distribution of data for each robot and the optimal paths corresponding to this offline solution are shown in Fig. 55.

After the initial offline solution, each agent then solves its own co-optimization problem online in a fashion similar to the ones described earlier and the results are shown in Fig. 56. After every 10 seconds, each agent performs its own channel prediction and optimization and updates its control policy as shown by the deviation

Figure 55: Six agents starting at the same point and moving towards their respective destinations. The data is distributed between different agents based on the channel qualities along the optimal paths. The peaks are the regions of bad channel quality an vice versa. The robots avoid regions of bad channel quality.

of each robot from its initial trajectory. The robot learns about the channel as it gathers more channel information and makes a more informed decision on its motion and transmission. This gives one example of extension of the result and several other are possible. Details of our work on single and multi-agent online co-optimization can be found in [1].

In this chapter and the previous, we considered the spectral efficiency to be a continuous variable. Most of the modulation schemes for variable power transmission however require the spectral efficiency to be discrete and thus the problem of co-optimization is hybrid not only in its objective function but also in the system dynamics and solving this problem in the framework of hybrid optimal control subject to various constraints is the topic of the next chapter.

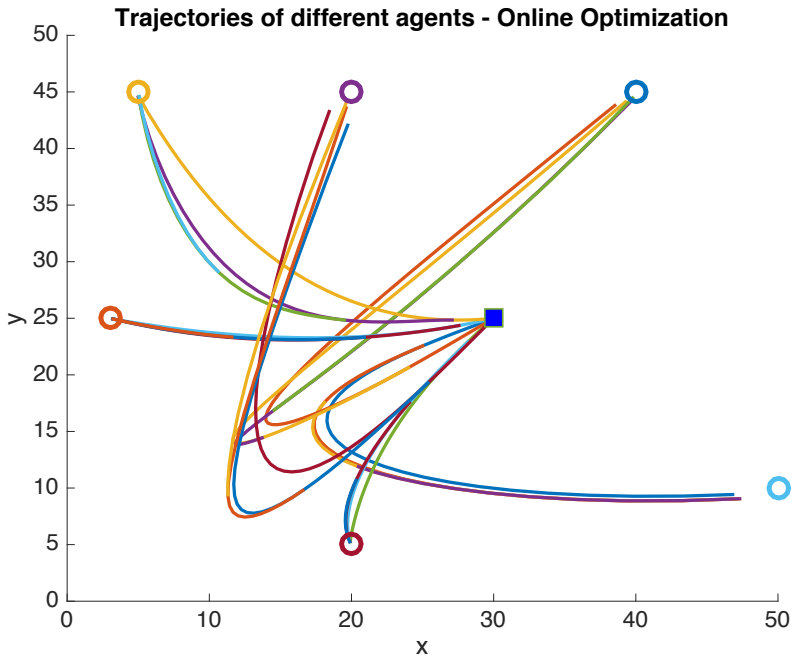


Figure 56: After initial planning, the robots perform channel prediction and optimization every 10 seconds. After initial data distribution each agent performs its own online optimization. The paths close together show the change in path planning for each individual robot as it learns about the channel and re-plan.

CHAPTER VIII

CO-OPTIMIZATION PROBLEM WITH HYBRID DYNAMICS

In the previous chapter, we dealt with the problem of motion and communication balancing using an optimal control approach assuming continuous controls for robot's acceleration u and spectral efficiency R. The hybrid nature of the problem thus resulted from being hybrid in the objective function only. The spectral efficiency R however takes on discrete values as explained below and hence the problem of Hybrid Co-optimization is what is considered next in this chapter. In fact, the spectral efficiency cannot switch instantly and as such, the hybrid co-optimization problem also exhibits dwell time constraints and thus amenable to our dwell time framework and this is the common thread between the two main topics of this thesis.

MQAM (M-ary Quadrature Amplitude Modulation) is a common digital modulation technique that allows for a variable rate and variable power transmission and this is the scheme which we have assumed in [6, 45]. The constellation diagram of Fig. 57 shows an example of 16QAM for transmitting a total of 16 possible signals and where each transmitted signal carries 4 bits of information. In general, if there is a total of M possible signals that need to be transmitted, one requires $\log_2 M$ bits. The constellation size is thus usually designed to be a power of 2. As mentioned before, the spectral efficiency is a measure of information rate over the given bandwidth, so if we assume that it takes T seconds to transmit a signal, the information rate is $\log_2(\frac{M}{T})$ and this normalized by the available bandwidth $(\frac{1}{T})$ gives the spectral efficiency to be $R = \log_2(M)$. Since usually $M = 2^N$ with $N = 0, 1, 2...$, we have thus $R = N$, with N being zero implying nothing is transmitted. This discussion

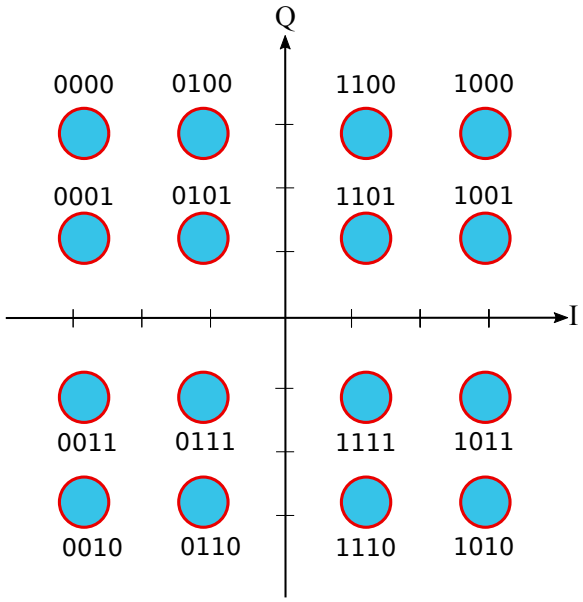


Figure 57: A constellation diagram for 16 QAM. It is possible to transmit more bits per symbol by moving to a higher order constellation, like 64 QAM or 256 QAM, which are also the most common forms of QAM.

motivates us to solve the co-optimization problem with R as a discrete variable and thus the problem of co-optimization considered in the previous chapters becomes a truly hybrid optimal control problem. In this chapter, we present the formulation or our problem as a switched mode constrained hybrid optimal control problem and then present a method for solving the problem. To focus on the essence of the problem, we consider the co-optimization problem of [6] considered in chapter 7, where a single robot is tasked to transmit a given number of bits along a fixed path in a given time horizon an the problem.

8.1 Problem Formulation

Consider the problem of a robot tasked to transmit a given number of bits required to traversed a predetermined path in a fixed time interval $[t_0, t_f]$. The channel quality at any point along the is estimated based on few measurements a priori and the robot has to optimized its acceleration and spectral efficiency as to minimize a joint cost defined on the motion and communication energy while satisfying constraints

on control inputs and on the state. The control variable acceleration is continuous but the control variable spectral efficiency is discrete that takes on integer values assuming MQAM modulation scheme. We use the same model for motion energy, communication energy, system dynamics and channel estimation as in chapter 7 and the meanings of the different variables remain the same. The co-optimization problem is then to minimize the cost functional

$$J = \int_0^{t_f} \left(\frac{2^{R(t)} - 1}{K} s(x_1(t)) + \gamma(k_1 u(t)^2 + k_2 x_2(t)^2 + k_3 x_2(t) + k_4 + k_5 u(t) + k_6 u(t) x_2(t)) \right) dt \tag{118}$$

subject to the dynamical and state constraints

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \qquad x_1(0) = a, \quad x_1(t_f) = b, \\ \dot{x}_2(t) &= u(t), \qquad x_2(0) = 0, \quad x_2(t_f) = c, \\ \dot{x}_3(t) &= R(t), \qquad x_3(0) = 0, \quad x_3(t_f) = d \end{aligned} \tag{119}$$

and constraints on the input

$$\begin{aligned} -u_{\max} \leq u(t) \leq u_{\max}, \\ R(t) \in \mathcal{Q} = \{0, 1, 2, 3, 4, 5, 6, 7\} \end{aligned} \tag{120}$$

8.2 Hybrid Dynamics

We can reformulate the problem by considering the discrete input $R(t)$ as a switching signal that chooses between different modes, corresponding to discrete value of R, while treating u as a continuous input to each subsystem. We can represent the dynamics (119) as hybrid switched system where the dynamics of each mode are given by

$$\dot{x} = f_i(x, u) = Ax + Bu + Cr_i \tag{121}$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

For each $r_i \in \mathcal{Q}$, we get one subsystem. So in total we have then 8 subsystems or modes corresponding to 8 different values of spectral efficiency and thus the spectral efficiency can be viewed as the switching signal $v(t)$ that performs mode selection, while u is the continuous input to each subsystem that belongs to $\mathcal{U} = [-1, 1]$ and thus the dynamics in compact form again are

$$\dot{x}(t) = f_{R(t)}\big(x(t), u(t)\big), \quad x(0) = x_0 \tag{122}$$

and where we have purposely left out the terminal constraints on the state above, since we are going to add them to cost via penalty terms for computational purposes. The cost function is also dependent on the mode we are in and thus

$$\bar{J} = \int_0^{t_f} L_{R(t)}\big(x(t), u(t)\big) + \phi(x(t_f)) \tag{123}$$

where the cost function L is now mode dependent

$$L_{r_i}(x, u) = \frac{2^{r_i} - 1}{K} s(x_1(t)) + \gamma(k_1 u(t)^2 + k_2 x_2(t)^2 + k_3 x_2(t) + k_4 + k_5 u(t) + k_6 u(t) x_2(t))$$

and the terminal cost accommodates for the final state constraints as

$$\phi(x(t_f)) = C_1 ||x_1(t_f) - b||^2 + C_2 ||x_2(t_f) - c||^2 + C_3 ||x_3(t_f) - d||^2.$$

So we have a controlled hybrid dynamical system and we are looking towards case-3 of *Full Optimization* introduced in section 2.2, where we are seeking optimization over the schedule $R(t)$ (spectral efficiency) as well as the continuous input $u(t)$.

Hybrid Co-Optimization Problem:

The Hybrid Co-optimization problem then is

$$\min_{\substack{R \in \mathcal{Q} \\ u \in \mathcal{U}}} \quad \bar{J}(R, u) \tag{124}$$

subject to dynamical and initial condition constraints (122).

8.3 Computing the Optimal Controls

To compute the optimal solution, we solve a relaxed version of the problem and then project the solution onto the switching space. For this, we consider the convex combination of all vector fields. The corresponding embedded system dynamics are then give by

$$\dot{x}_e = \sum_{i=1}^{8} \alpha_i \big(Ax + Bu_i + Cr_i \big) \tag{125}$$

subject to the constraints

$$\sum_{i=1}^{8} \alpha_i = 1, \qquad \alpha_i \geq 0$$

Similarly, the embedded cost is obtained as a convex combination of the cost functionals and is given by

$$J_e = \int_0^{t_f} \sum_{i=1}^{8} \alpha_i \Bigg(\frac{2^{r_i(t)} - 1}{K} s(x_1(t)) + \gamma (k_1 u_i(t)^2 + k_2 x_2(t)^2 + k_3 x_2(t) + k_4 + k_5 u_i(t) + k_6 u_i(t) x_2(t)) \Bigg) dt \tag{126}$$

Including the terminal constraints as part of this cost, we get the co-state equations

$$\begin{aligned}
\dot{p}_1 &= -\sum_{i=1}^{4} \alpha_i \left(\frac{2^{r_i(t)} - 1}{K} \right) \frac{\partial s(x_1)}{\partial x_1}, && p_1(t_f) = 2C_1(x_1(t_f) - b), \\
\dot{p}_2 &= -p_1 - 2\gamma k_2 x_2 - \gamma k_3 - \gamma k_6 \sum_{i=1}^{8} \alpha_i u_i, && p_2(t_f) = 2C_2(x_2(t_f) - c), \\
\dot{p}_3 &= 0, && p_3(t_f) = 2C_3(x_3(t_f) - d),
\end{aligned} \tag{127}$$

For each mode, the optimal u_i is given by

$$u_i^\star = \begin{cases} -\dfrac{(p_2 + \gamma(k_5 + k_6 x_2))}{2\gamma k_1} & \text{if } \left| \dfrac{(p_2 + k_5 + k_6 x_2)}{2k_1} \right| \leq u_{\max}, \\ u_{\max} & \text{if } -\dfrac{(p_2 + \gamma(k_5 + k_6 x_2))}{2\gamma k_1} > u_{\max}, \\ -u_{\max} & \text{if } -\dfrac{(p_2 + \gamma(k_5 + k_6 x_2))}{2\gamma k_1} < -u_{\max}, \end{cases} \tag{128}$$

Let $(j(t), u_j^*(t))$ be the optimal controls and where $R^*(t) = j(t) - 1$. Then we can write the hamiltonian associated with this control as

$$H^*(x, u, p) = p^T f_j(x, u_j^*) + L_j(x, u_j^*) \tag{129}$$

whereas the Hamiltonian associated the embedded system is given by

$$H(x, u, p) = p^T \sum_{i=1}^{8} \alpha_i f_i(x, u_i) + \sum_{i=1}^{8} \alpha_i L_i(x, u_i). \tag{130}$$

The optimality functions is then

$$\theta = \int_0^{t_f} (H^*(x, u, p) - H(x, u, p))dt \tag{131}$$

which is always less than or equal to zero, and it being zero signifies the optimality conditions are met. We then use the Hamiltonian algorithm presented in chapter 6 which employs the Armijo step size rule to compute the optimal controls. The controls at any given stage of iterations are updated as the convex combinations of existing vector field and the vector field associated with the minimizer $(j,\, u_j^*)$ of the Hamiltonian. So we have

$$f_{next} = (1 - \lambda) f_{current} + \lambda f_{optimal} \tag{132}$$

where λ is the step size chosen according to Armijo rule, so that the algorithm converges to the optimal solution and f is used to denote the vector fields. If we plugin the values and perform some algebra, we get a rule for updating the optimal controls

and the weights associated with each mode as follows

$$
\begin{aligned}
& f_{next} \\
& = (1-\lambda)\sum_{i=1}^{8}\alpha_i f_i(x,u_i) + \lambda f_j(x,u_j^*) \\
& = \sum_{i=1,i\neq j}^{8}(1-\lambda)\alpha_i f_i(x,u_i) + (1-\lambda)\alpha_j f(x,u_j) + \lambda f_j(x,u_j^*) \\
& = \sum_{i=1,i\neq j}^{8}(1-\lambda)\alpha_i f_i(x,u_i) + ((1-\lambda)\alpha_j+\lambda)\left[\frac{(1-\lambda)\alpha_j}{(1-\lambda)\alpha_j+\lambda}f(x,u_j) + \frac{\lambda}{(1-\lambda)\alpha_j+\lambda}f_j(x,u_j^*)\right] \\
& = \sum_{i=1,i\neq j}^{8}(1-\lambda)\alpha_i f_i(x,u_i) + ((1-\lambda)\alpha_j+\lambda)\left[f_j\left(x, \frac{(1-\lambda)\alpha_j}{(1-\lambda)\alpha_j+\lambda}u_j + \frac{\lambda}{(1-\lambda)\alpha_j+\lambda}u_j^*\right)\right] \\
& = \sum_{i=1}^{8}\beta_i f_i(x,w_i) \qquad (133)
\end{aligned}
$$

with

$$
\begin{aligned}
\beta_i &= (1-\lambda)\alpha_i + \lambda, \\
w_i &= \frac{(1-\lambda)\alpha_i}{(1-\lambda)\alpha_i+\lambda}u_i + \frac{\lambda}{(1-\lambda)\alpha_i+\lambda}u_i^* \qquad (134)
\end{aligned}
$$

for $i = j$ and

$$
\begin{aligned}
\beta_i &= (1-\lambda)\alpha_i, \\
w_i &= u_i \qquad (135)
\end{aligned}
$$

otherwise i.e $i \neq j$. In the above simplification, we have used the fact that the vector fields f_i are affine in the controls u_i. This gives the update rule for the weights on the modes and the continuous controls at each step of the iteration.

The optimal solution is computed in the relaxed space and we somehow need to go back to the switching space by using some kind of projection. One common approach is to use schemes like pulse width modulation [38]. The switching between different modes is then approximated by assigning each mode a duration of pulse in accordance with its weights at the beginning of the pulse, and during each mode, the respective optimal $u_i^*(t)$ is applied. This will be explained next in a an application example.

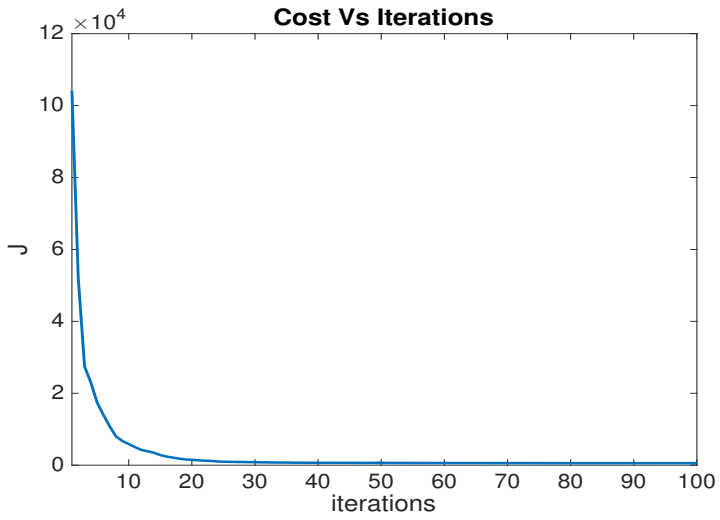


Figure 58: Plot of cost vs. iterations for the hybrid co-optimization problem. The cost decreases rapidly during the first few iterations.

8.4 Simulation Results

For solving the hybrid co-optimization problem, we consider the same problem of chapter 6 with the same start and ending points and constraints on the inputs. The channel data and parameters are chosen to be the same as before and the same goes for Armijo parameters and the choice of the step size for solving the differential equations. The results of our simulation are shown in Fig. 58 - 62 which we briefly explain next. The plot for cost (J) versus iteration count is depicted in Fig. 58, showing a rapid decrease in cost during the first few iterations of the algorithm. The cost decreases from the initial value of 1.0408×10^5 to 557 in 100 iterations. The cost shows a rapid descent during the initial steps of the algorithm thanks to the Armijo step size rule. The final states are $x(t_f) = (19.55, 0.25, 99.93)$ very close to the desired final states of $x_f = (20, 0, 100)$ for the given choice of penalty terms.

Fig. 59 shows the channel quality and relaxed spectral efficiency ($R = \sum_{i=1}^{8} \alpha_i^*(t) r_i$) and the two are scaled for effective presentation. As can be seen, the spectral efficiency is high in the region with good channel quality, corresponding to smaller values of $s(x_1)$ and vice versa. Fig. 60 shows the relaxed acceleration ($u = \sum_{i=1}^{8} \alpha_i^*(t) u_i^*(t)$)

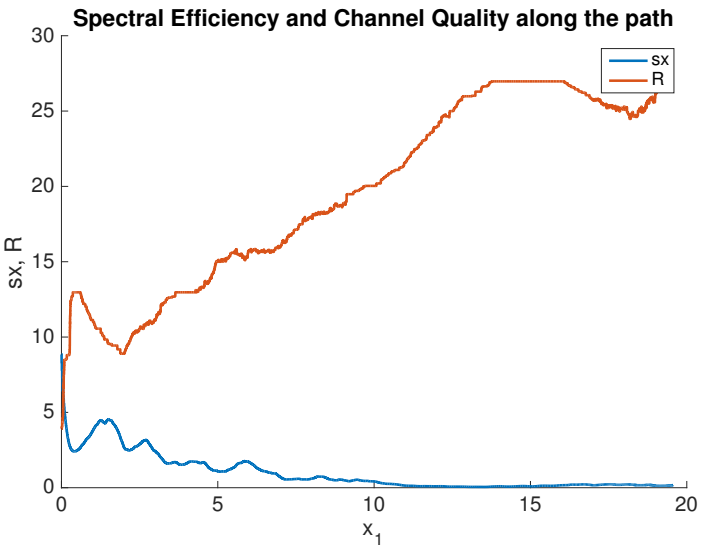


Figure 59: Spectral efficiency and channel quality at different points along the path. The spectral efficiency is high in regions of good channel quality and vice versa. The spectral efficiency and channel quality are both scaled for vivid depiction. The jitteriness is the result of this being the convex summation of weights on discrete spectral efficiencies to form the resultant relaxed spectral efficiency shown in this figure.

of the robot. The robot tries to quickly get out of the region of the bad channel quality and towards the end decelerate to bring the robot to a stop. All these results agree with our intuition.

While the above results give us an idea of the appropriate behaviors, we need to project our optimal controls onto the space of switching controls for implementation and this is what we consider next. The optimal weights for each mode are shown in Fig. 61. Because of the high transmission demand, the robot assumes the higher mode, corresponding to large values of spectral efficiency and this is indicated by the corresponding large weights on the spectral efficiency. To be practically useful, we however need somehow to compute the discrete spectral efficiency utilizing the information of different weights on the modes. This can be done using pulse width modulation (PWM) as in [38]. For the purpose of depiction, we choose a large duration of 2 seconds for the pulse and portions of this pulse are assigned to different

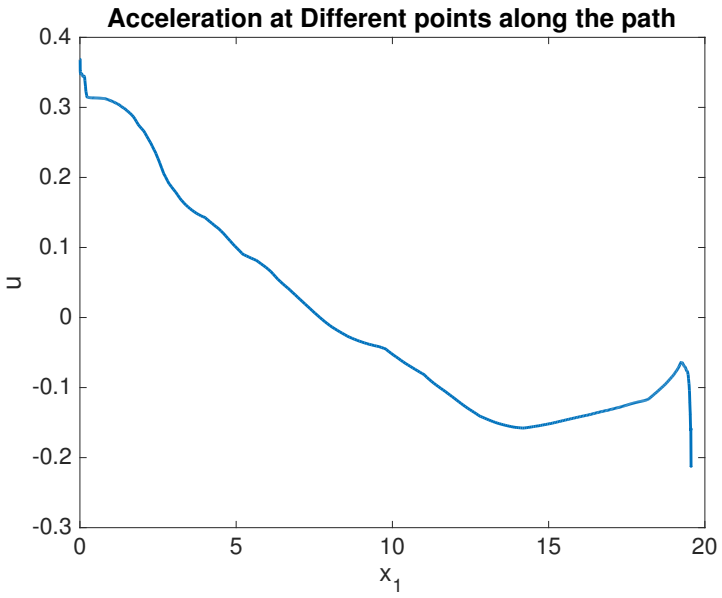


Figure 60: Acceleration at different points along the path. The channel quality is relatively bad near the origin and thus the robot accelerates to get out of this region. The robot makes a deceleration towards the end to bring the robot to a stop, a condition specified by terminal constraints on the problem.

modes (and hence spectral efficiencies) in according with their corresponding weights at the beginning of the pulse. The results are show in Fig. 62. The value of the cost using this discrete $R(t)$ and continuous inputs $u_i^*(t)$ for each mode is 577, compared to 557 for the relaxed spectral efficiency. This difference can however easily be reduced by choosing small duration for the pulse and thus switching more rapidly between the modes to closely approximate the relaxed spectral efficiency using the discrete modes.

Towards the end of this chapter, we want to point out that the spectral efficiency cannot switch instantly and thus the hybrid co-optimization problem is also subject to our dwell time constraints and thus amenable to our framework. This is the case where the dwell time is very small, few milliseconds, as opposed to the dwell time constraints of days for the optimal pesticide scheduling problem and thus we have a range of hybrid optimal control problems between the two opposite extremes of the spectrum that can be addressed using the dwell time framework presented in chapters

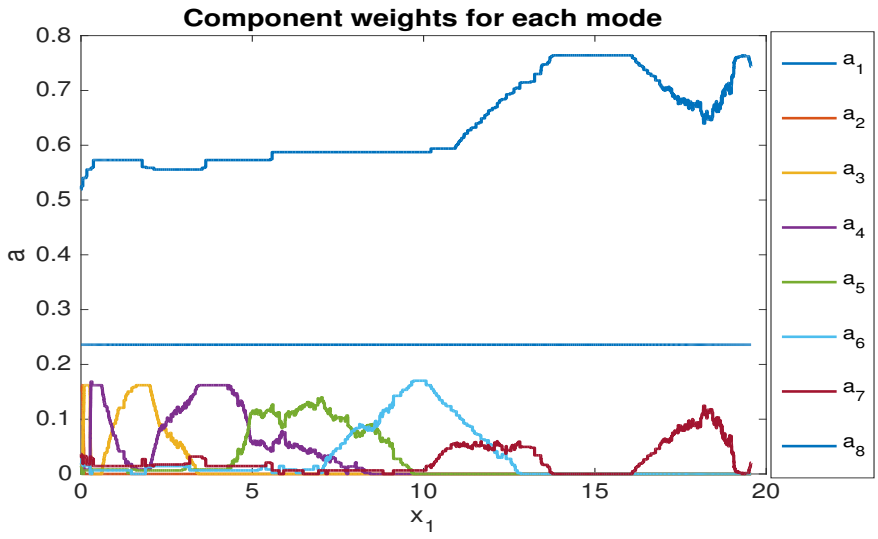


Figure 61: Weights associated with each mode at different points in time. The weight on mode 8 is high all the time due to high transmission demand set for this simulation problem.

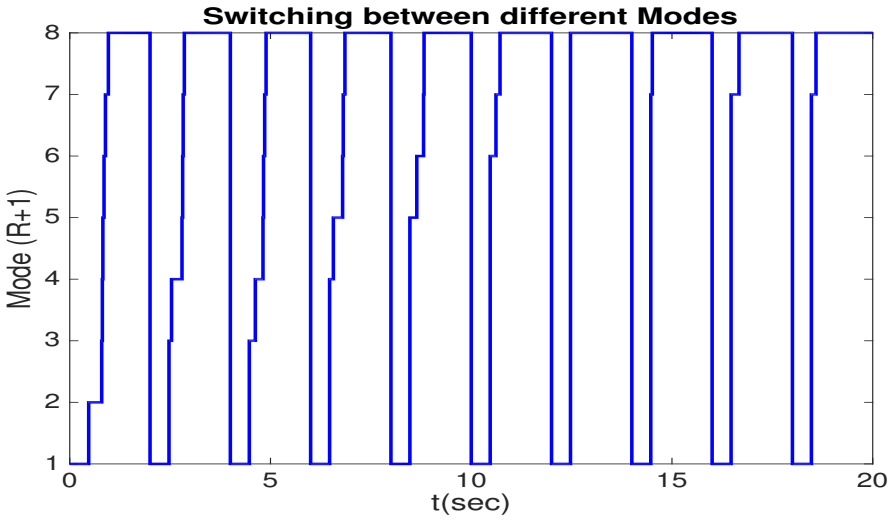


Figure 62: The use of pulse width modulation scheme to arrive at the switching signal. A fixed pulse length is assumed and then portions of this pulse are assigned to different modes according to their respective weighs at the beginning of the pulse. In this realization, the pulse duration is chosen to be 2 seconds for vivid depiction.

3-5. We presented one interesting example of optimal pesticide scheduling in chapter 5 to demonstrate the application of our methodology.

The Hamiltonian algorithm and for that matter other optimization methods in general, have problem solving optimal control problems over large time horizon. In the next chapter we present an optimal control method for solving complex problems over large time horizons and consider its application to solve the problem of co-optimization in large multi-agent power aware networks as a specific example.

CHAPTER IX

MULTIPLE SHOOTING TECHNIQUE FOR OPTIMAL AND HYBRID OPTIMAL CONTROL PROBLEMS

In this chapter, we introduce a multiple shooting based technique for dealing with complex optimal and hybrid optimal control problems over large horizon. The problem surfaces from the fact that if the states are stable, the costates are unstable and vice versa and which results in numerical issues and is a reason for slow convergence of our Hamiltonian algorithm presented in chapter 6 when dealing with problems over large horizon and hence the need for a computational technique that can deal with such issues. Below we present a multiple shooting based technique to tackle this issue in an effective manner and while the method presented is quite general, we will specifically consider the problem of solving multi-agent complex optimal control problems in power aware networks. Towards the end of this chapter, we highlight how this approach can be implemented for solving a class of hybrid optimal control problems. The work presented in this chapter also appeared in [7].

The class of optimal control problems considered for the application of our computational technique have the property that the optimal control is expressed (pointwise) via feedback in terms of the state and costate (adjoint) variables. The technique, combining gradient descent with a multiple shooting method, is tested on a power-aware problem in a fifty-agent network and it yields convergence towards a minimum.

Consider the following general Bolza problem with pointwise input constraints: The state equation is

$$\dot{x}(t) = f(x(t), u(t)), \tag{136}$$

where $x(t) \in R^n$, $u(t) \in R^k$, and $f : R^n \times R^k \to R^n$ is continuously differentiable in x

and locally Lipschitz continuous in u. The initial time is $t = 0$, and the initial state $x(0) := x_0$ and final time $t_f \in (0, \infty)$ are given and fixed, and suppose that there exists a solution $\{x(t)\}_{t=0}^{t_f}$ for every piecewise-continuous control function $u(t)$, $t \in [0, t_f]$. Let $L : R^n \times R^k \to R$ be a function that is continuously differentiable in x and locally Lipschitz continuous in u, and let $\phi : R^n \to R$ be a continuously-differentiable function; define the cost functional J by

$$J = \int_0^{t_f} L(x(t), u(t))dt + \phi(x(t_f)). \tag{137}$$

Given a compact and convex set $U \subset R^k$, the optimal control problem is to minimize J over all piecewise-continuous functions $u : [0, t_f] \to U$. A control function satisfying these constraints is said to be admissible.

The maximum principle asserts that if $\{u(t)\}$ is an optimal control then the Hamiltonian is minimized at the value of the control ([58], and [17]). That is, if $\{u(t)\}$ is an optimal control, and $\{x(t)\}$ and $\{p(t)\}$ are its associated state and costate trajectories, then at every $t \in [0, t_f]$, $u(t) \in argmin\{H(x(t), w, p(t)) : w \in U\}$, where

$$H(x(t), u(t), p(t)) := p^\top f(x(t), u(t)) + L(x(t), u(t))$$

is the Hamiltonian function. Suppose now that there exists a Lebesgue-measurable function $h : R^n \times R^n \to R^k$ such that, for every $x \in R^n$ and $p \in R^n$,

$$u := h(x(t), p(t)) \in argmin\{H(x(t), w(t), p(t)) : w \in U\}. \tag{138}$$

The function h provides a feedback law for the optimal control, and the optimal control problem becomes that of solving the following two-point boundary value problem: Define the *extended state* by $z := [x, p]^\top \in R^{2n}$, and the corresponding (extended) state equation via [1]

[1]The vector fields can be discontinuous in the variable z due to constraints on u. As long as the condition of transversality holds at the surface of discontinuity, this does not affect the multiple shooting technique discussed in this chapter, since we only require continuity with respect to initial conditions; see [29]. This allows us to consider the extension of our results for a class of hybrid optimal control problems discussed towards the end of this chapter.

$$\dot{z} = F(z) := \begin{bmatrix} f\big(x, h(x,p)\big) \\ -\Big(\frac{\partial f}{\partial x}\big(x, h(x,p)\big)\Big)^{\top} p - \Big(\frac{\partial L}{\partial x}\big(x, h(x,p)\big)\Big)^{\top} \end{bmatrix}, \tag{139}$$

with the notational dependence on t omitted here and in the sequel. The initial condition on x is $x(0) = x_0$ while $p(0)$ is free; $x(t_f)$ is free while $p(t_f) = \nabla\phi(x(t_f)$. The problem is to compute $p(0) := p_0$ such that $p(t_f) = \nabla\phi(x(t_f))$.

Various algorithmic techniques have been proposed for solving this problem, commonly known as shooting methods; see survey papers by [60] and [14] and references therein. Generally they can be fraught with numerical stability problems when the final time t_f is large. To get around this issue, the shooting methods can be partitioned by dividing the time interval $[0, t_f]$ into several subintervals, and running a shooting method at each subinterval while seeming the resulting extended state at their boundaries via penalty terms as explained below. These enhanced techniques, called multiple shooting methods, often perform better than the non-concatenated techniques ([60]).

We cast the problem of minimizing the cost (137) subject to the dynamical constraints (136) and initial condition constraints, as an equivalent optimal control problem of minimizing an alternative cost defined on the terminal conditions on extended state variable z subject to the dynamical constraints (139). The variable of this cost is the initial condition of the extended state variable. To get around the aforementioned stability issues, the technique of multiple shooting is used and this is explained in this chapter. The problem is then solved using gradient descent algorithm where the gradient of the new cost is used as a descent direction and Armijo rule ([9]) is used to compute the step-size in the descent direction. The optimal initial condition gives the optimal state and costate trajectories via (139) and the optimal control is then computed using (138). To demonstrate the effectiveness of our method, we consider the application of the proposed technique to a power aware problem that

involves co-optimization of mobility and communication power.

The rest of this chapter is organized as follows. In section 1, we survey the principle behind multiple shooting methods and explain how we propose to use it in conjunction with a gradient-descent technique to solve the optimal control problem. Section 2 defines the power aware problem, and section 3 presents the numerical results. Section 4 discusses extension to a class of continuously-controlled hybrid optimal control problems. Section 5 concludes the chapter.

9.1 Multiple Shooting: Problem definition and gradient descent approach

In this section, we transcribe the boundary value problem (139) into an optimization problem over the initial costate variable $p_0 := p(0)$. Since the initial condition x_0 is fixed and we need to select p_0 so as to satisfy the terminal condition on the costate, we introduce a new cost functional

$$\tilde{J}(p_0) = \frac{1}{2}||p(t_f) - \nabla\phi(x(t_f))||^2 \tag{140}$$

and define the new optimization problem as

$$\min_{p_0} \tilde{J}(p_0) \tag{141}$$

subject to Eq. (139) and initial condition $x(0) = x_0$. Let us denote the costate associated with the dynamics in Eq. (139) by $\lambda := [\lambda_1,\ \lambda_2]^\top$, where $\lambda_1 \in R^n$ and $\lambda_2 \in R^n$. Recall the costate equation associated with Eq. (136), namely

$$\dot{p} = -\Big(\frac{\partial f}{\partial x}\big(x,u\big)\Big)^\top p - \Big(\frac{\partial L}{\partial x}\big(x,u\big)\Big)^\top. \tag{142}$$

Substitute for u in it from Eq. (138), and denote its Right-Hand Side (RHS) by $g(x,p)$. Similarly, substitute for u in Eq. (136) from Eq. (138), and denote the RHS of Eq. (136) by $f(x,p)$. Then it is readily seen that λ_1 and λ_2 are defined by the

following equation,

$$\dot{\lambda}_1 = -\Big(\frac{\partial f}{\partial x}\Big)^\top \lambda_1 - \Big(\frac{\partial g}{\partial x}\Big)^\top \lambda_2$$
$$\dot{\lambda}_2 = -\Big(\frac{\partial f}{\partial p}\Big)^\top \lambda_1 - \Big(\frac{\partial g}{\partial p}\Big)^\top \lambda_2, \tag{143}$$

with the boundary conditions

$$\lambda_1(t_f) = 0, \qquad \lambda_2(t_f) = p(t_f) - \nabla\phi(x(t_f)).$$

Now the gradient of cost $\tilde{J}(p_0)$ in (141) is

$$\frac{d\tilde{J}}{dp_0} = \lambda_2(0); \tag{144}$$

see [17].

To distinguish λ from the original costate variable p, we call λ as the meta costate. Similarly we will refer to the cost in Eq. (140) as meta cost to differentiate it from the original cost in Eq. (137).

As mentioned in the introduction, optimizing over single initial condition using shooting method can have stability issues. We next discuss the more general multiple shooting method, that introduces auxiliary shooting points along the costate trajectory at different time instants. We use the multiple shooting method in the following manner. Partition the interval $[0, t_f]$ at given time instances t_0, t_1, ..., t_m such that

$$0 = t_0 < t_1 < t_2, \ldots < t_m < t_f.$$

Defining $\bar{p}_0 := (p_{0,0}, \ldots, p_{0,m})^\top \in R^{m+1}$ to be control parameters that act as initial costates at these respective time instants, consider the problem of minimizing the surrogate cost function

$$\bar{J}(\bar{p}_0) := \frac{K}{2}\sum_{i=1}^{m}\big(||p(t_i) - p_{0,i})||^2\big) + \frac{1}{2}||p(t_f) - \nabla\phi(x(t_f))||^2. \tag{145}$$

The costate p is continuous, and hence the term with K in this equation acts to drive $p(t_i)$ towards $p_{0,i}$, which is the initial costate for the $(i+1)^{th}$ shooting segment.

This results in smoothing out the jumps in costate trajectory at shooting points and hence the final costate is continuous. The gradients of this cost with respect to the initial costates are similar to the one in Eq. (144) and are given by

$$\frac{d\bar{J}}{dp_{0,i}} = \lambda_2(t_i), \quad i = 0, 1 \ldots, m. \tag{146}$$

Furthermore, the meta costate equations Eq. (143) remain the same except for the the additional constraints at the intermediary shooting points,

$$\lambda_2(t_i) = K(p(t_i) - p_{0,i}), \quad i = 1, \ldots, m. \tag{147}$$

The terminal constraints on the meta costates λ_1 and λ_2 remain the same as in Eq. (143). The multiple shooting method can be seen as a dual of the switch time optimization problem, where the switch times are now fixed but we are trying to move the initial conditions at these switching instants in the vertical direction to satisfy the constraints on the co-states in this case.

Finally, we use the steepest-descent algorithm presented in [57] to solve the boundary value problem (139) and hence compute the optimal control (138). The descent direction used in the algorithm is given by the gradient formula (146) and the step size in the descent direction is chosen according to Armijo rule ([9]). To demonstrate the effectiveness of this method, we next consider its application to an example.

9.2 Application to a Power-aware Problem

We consider an information-relay system where data is to be transferred from a fixed source to a fixed destination over a mobile relay network. The network is comprised of a given number of mobile relay nodes (agents) which, for simplicity of argument, are assumed to be arranged in a fixed logical order. Thus, denoting the agents by A_i, $i = 1, \ldots, N$, and denoting the source and destination by A_0 and A_{N+1}, respectively, the data is being transmitted in series between A_i and A_{i+1}, $i = 0, \ldots, N$.

Suppose that at time $t = 0$, the agents are instructed by a high-level (supervisory) controller to transmit data between a source and a destination at a given rate, for a given amount of time $t \in [0, t_f]$. Starting at time $t = 0$, the agents start moving while transmitting, and their common objective is to balance the energies expanded on transmission and motion by minimizing a weighted sum of the two types of energy. Such a problem falls under the category of *power-aware optimization*, and variants of it have been the focus of recent and current research (see e.g. [32], [84], [83], [43], [6] etc.). This chapter considers a simplified model since its purpose is to provide a proof of concept for our algorithmic technique, and investigations of more realistic models is forthcoming (see the Conclusions section).

Let $x_i \in R^2$ and $u_i \in R^2$ denote the planar position and velocity of A_i, $i = 1, \ldots, N$, and let the state and input of the system be $x := (x_1^\top, \ldots, x_N^\top)^\top \in R^{2N}$ and $u := (u_1^\top, \ldots, u_N^\top)^\top \in R^{2N}$. Suppose that the transmission power on a link of distance z is proportional to z^2, and the power required for a relay's motion is proportional to the square of its speed. Thus, the state equation is

$$\dot{x}_i = u_i, \quad i = 1, \ldots, N, \tag{148}$$

for a given initial condition $x_i(0) \in R^2$, and the cost functional, J, has the form

$$J = \frac{1}{2} \int_0^T \left(\sum_{i=0}^{N} ||x_{i+1} - x_i||^2 + C \sum_{i=1}^{N} ||u_i||^2 \right) dt, \tag{149}$$

where $C > 0$ is a given constant, and x_0 and x_{N+1} denote the (constant) positions of the source and destination, respectively. Let u_{max} be a given upper bound on the speed of the agents. Then the optimal control problem is to minimize J subject to the dynamics in (148) and the pointwise constraints $||u_i|| \leq u_{max}$.

The costate $p \in R^{2N}$ also can be decomposed into N two-dimensional vectors p_i, $i = 1, \ldots, N$, namely $p = (p_1^\top, \ldots, p_N^\top)^\top$, and it is readily seen that the costate equation has the form

$$\dot{p}_i = x_{i-1} - 2x_i + x_{i+1}, \qquad i = 1, \ldots, N, \tag{150}$$

with the boundary conditions $p_i(t_f) = 0$. Consequently the Hamiltonian has the form

$$H(x, u, p) = \sum_{i=1}^{N} p_i^\top u_i + \frac{1}{2}\left(\sum_{i=0}^{N} ||x_{i+1} - x_i||^2 + \sum_{i=1}^{N} ||u_i||^2\right), \tag{151}$$

and its pointwise minimizer for given x and p is readily seen to be

$$u_i^* = \begin{cases} -p_i(t), & ||p_i|| \leq u_{max} \\ -u_{max}\frac{p_i(t)}{||p_i||}, & ||p_i|| > u_{max}. \end{cases} \tag{152}$$

Eq. (152) provides a feedback for Eq. (148) and Eq. (150) resulting in the two point boundary value problem of the form (139).

9.3 Simulation Results

We coded our algorithm in MATLAB, and ran it on a laptop with dual core Intel i5 2.7 GHz processor and 8GB RAM. Two problems were solved concerning two agents and fifty agents, respectively. The first problem serves to test the basic shooting method while the second one tests a multiple shooting technique on a large problem. In both the problems, the maximum velocity was $u_{max} = 1$, and the parameters K and C in Eq. (145) and Eq. (149) are set to 1.

9.3.1 Problem I: a two-agent network

The time horizon is $t_f = 10$ seconds, and the source and destination are in positions $O = (0, 0)$ and $S = (30, 0)$, respectively. The two agents are initialized to positions $x_1 = (2, 7)$ and $x_2 = (10, -5)$, respectively. The Armijo parameters are set to $\alpha = 10^{-4}$ and $\beta = 0.5$ (after some experimentation), and the problem is solved using a single shooting algorithm. The algorithm is run for 100 iterations. All differential equations are computed via the forward Euler method based on the time-difference $\Delta t = 0.01$s. The algorithm is terminated when the meta cost $\bar{J}$ (145) falls below value 1×10^{-4}. Fig. 63 shows the meta cost as a function of the iteration count k, and Fig. 64 shows the actual cost (149) as a function of k.

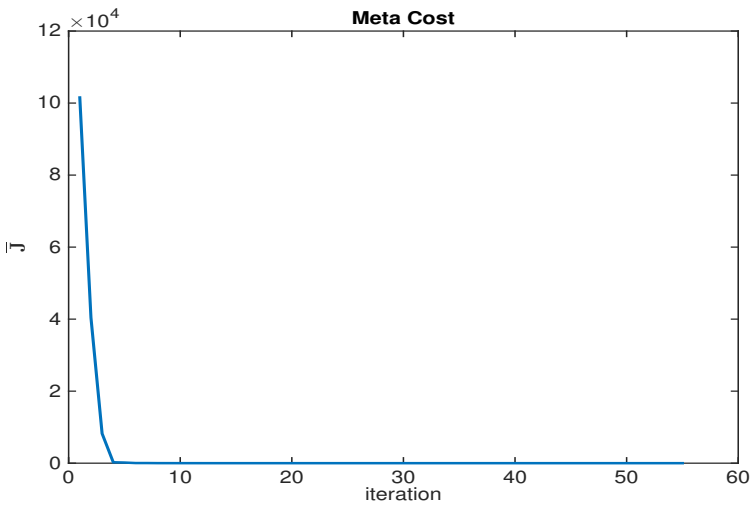


Figure 63: Plot of meta cost (140) vs. iterations. The algorithm reaches near the optimal solution in the first few iterations.

The figures indicate that much of the cost reduction occurs in first few iterations. In 15 iterations, the meta cost declines from its initial value of 1.0154×10^5 to 0.5673, while the actual cost declines from 1.2623×10^6 to 4.1612×10^5. The algorithm terminated after 55 iterations. The final values of meta cost and actual cost at the end of 55 iterations are 8×10^{-5} and 4.1601×10^5, respectively, which is close to the values at the iteration 15. The 55-iteration run took 7.79 seconds of cpu time. Fig. 65 depicts the evolution of the agents' trajectories from squares to circles, and we note that the agents approach the line segment joining the source and destination with which they form an equidistant arrangement. The large squares on the left and right denote the source and destination respectively.

9.3.2 Problem II: a fifty-agent network

The time-horizon is $t_f = 50$ seconds, and the source and destination are in positions $O = (0, 0)$ and $S = (50, 0)$, respectively. The agents are initialized to their positions as indicated by the squares in Fig. 68, where the (x,y) co-ordinates of each agent A_i are set to $(i, 5\sin(i))$ for $i = 1, \ldots, 50$. The Armijo parameters for the gradient descent algorithm are set to $\alpha = 5 \times 10^{-3}$ and $\beta = 0.5$ (for the definition of the

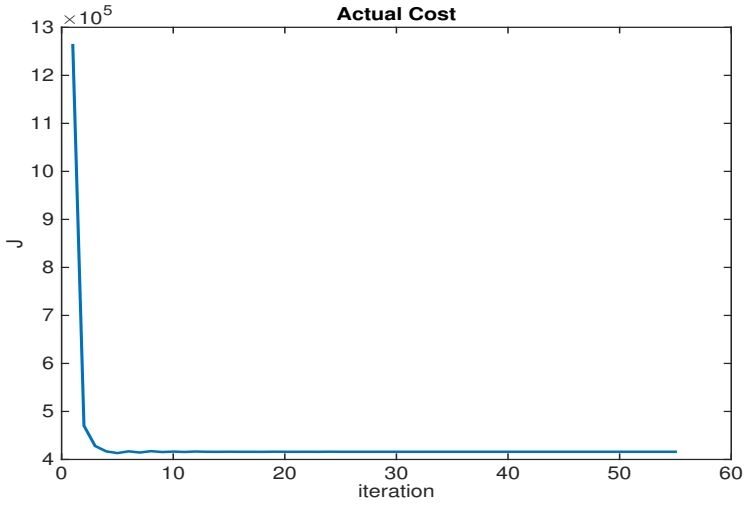


Figure 64: Plot of actual cost (137) vs. iterations. The actual cost shows the same pattern as the meta cost.

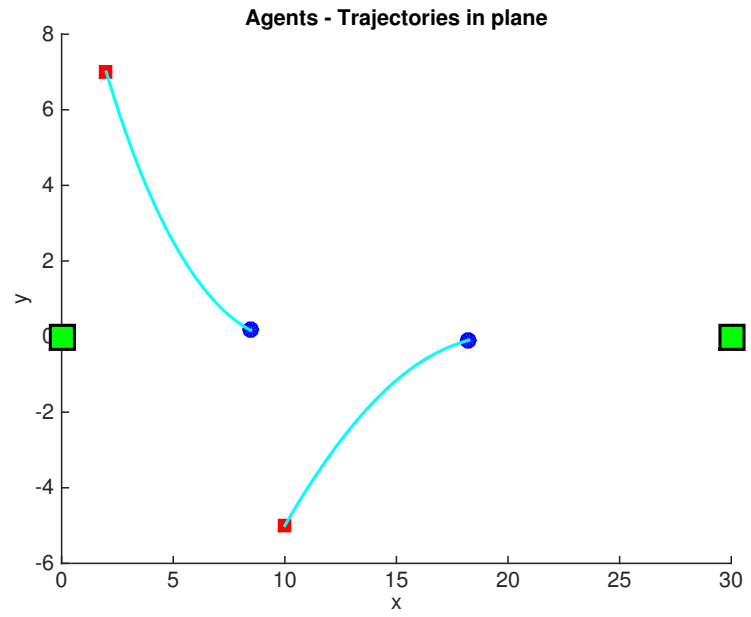


Figure 65: Path followed by agents in the plane. The big squares shown in green denote the fixed positions of source and destination while the small red squares are the initial positions of the mobile robots. The curves (shown in cyan) connecting the initial position of the robot to its final position (shown in blue circles) is the path taken by the robot in the plane.

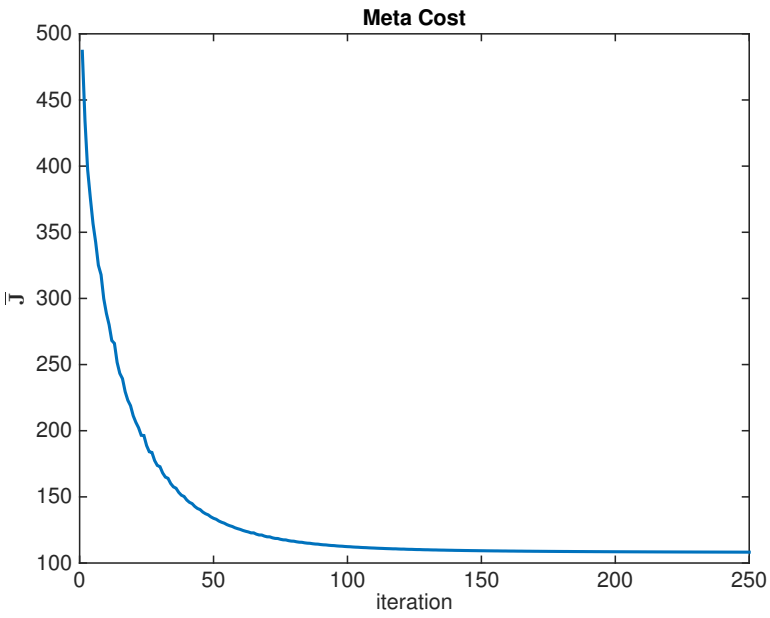


Figure 66: Plot of meta cost (140)Vs. iterations. The algorithm reaches near the optimal solution in about 100 iterations. This problem could not be handled by single shooting.

parameters α and β, see [57]), and the problem is solved using multiple shooting method. The time step used for integration was set to $\Delta t = 0.01$ seconds, and the multiple shooting was done every 0.05 seconds. The results for this case are shown in Fig. 66 - Fig. 68.

A 100-iterations run resulted in reductions in the meta-cost from 486.79 to 112.24, and in the actual cost from 5.6523×10^6 to 1.6018×10^6. After 250 iterations, the meta-cost and actual cost are 108.15 and 1.5527×10^6, respectively, and it took 446 seconds of cpu time.

Fig. 68 depicts the agents' motion from squares to circles throughout the algorithm's run. It is noted that the agents move towards the line joining source to destination (denoted by big squares to the left and right respectively), but they do not quite reach it due to the cost of motion.

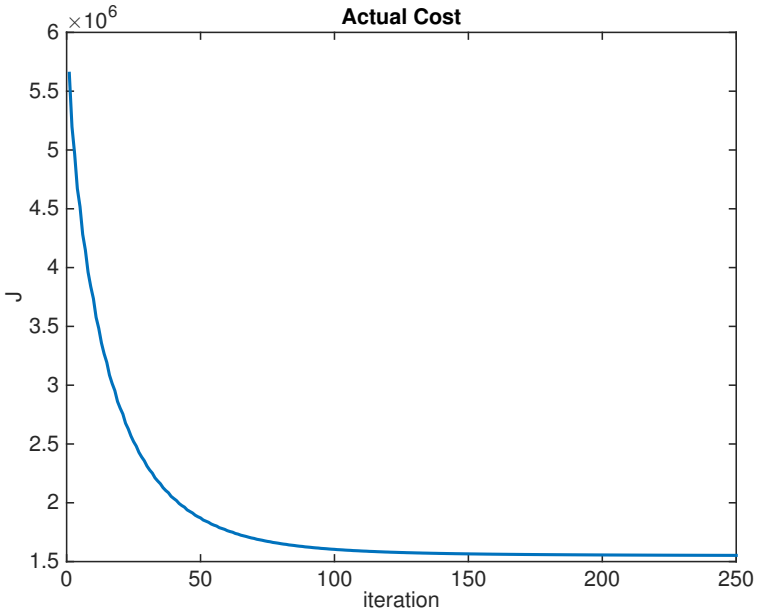


Figure 67: Plot of actual cost (137) vs. iterations. The actual cost shows the same pattern as the meta cost.

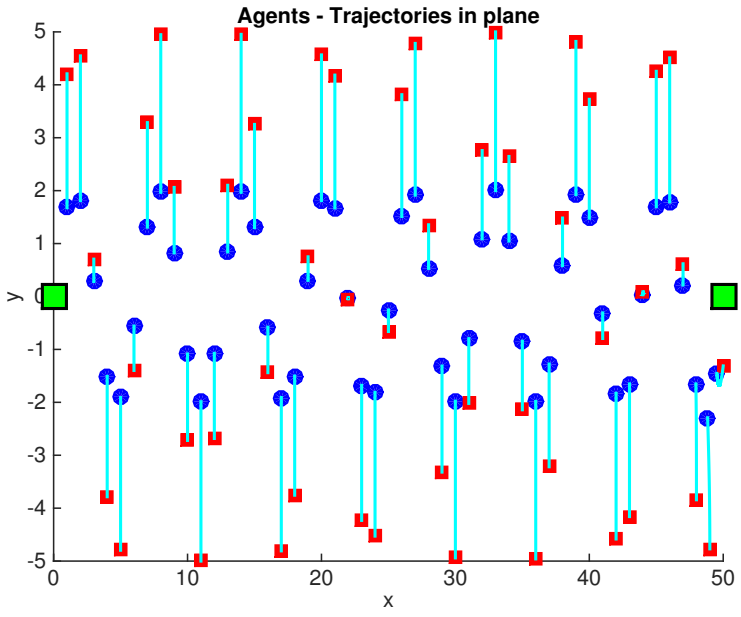


Figure 68: Path followed by agents in the plane. The big squares shown in green denote the fixed positions of source and destination while the small red squares are the initial positions of the mobile robots. The curves (shown in cyan) connecting the initial position of the robot to its final position (shown in blue circles) is the path taken by the robot in the plane.

9.4 Continuously-controlled Hybrid Optimal Control Problems

To demonstrate the methodology, we consider optimal control problems, however the results can be easily extended to hybrid optimal control problems. For example, hybrid optimal control problems with only continuous control u are amenable to the framework presented in this chapter. In this types of hybrid optimal control problem, the discrete switching control v is absent and the switching between different modes is dependent on state crossings the switching surfaces defined by constraints on the state variables. The control u alters the state trajectory and hence where the trajectory meets the switching surface and transitions to other mode. These types of problems can be easily addressed using the framework presented in this chapter except that there might be jumps in the co-state equations at the switching surface which needs to be taken into account. The gradient of cost w.r.t initial condition is however still continuous and this allows for perturbation of initial conditions using gradient descent based algorithm as described in this chapter. The extension to the case of *Full Optimization* however requires further investigation.

9.5 Conclusions

We considered a multiple shooting technique that can be very effective for solving a class of optimal control problems. Our approach uses multiple shooting method in conjunction with gradient descent for solving a two point boundary problem resulting from the optimal control problem. We considered the application of this technique to solve a power-aware problem of balancing motion with transmission energy in a 50-agent relay network. A remaining challenge is to develop a decentralized implementation that requires only little communications between neighboring agents.

CHAPTER X

CONCLUSION

The work presented in this thesis considers the optimal control of a class of hybrid dynamical systems under different type of constraints. Necessary conditions for optimality were developed for a class of hybrid optimal control problems that are constrained by how fast they can switch between different modes. We presented computational methods for computing solutions to these problems. The increase in discreteness of the problem due to the presence of dwell time constraints present significant computational challenge. A method was presented to use information on the insertion gradient, which are readily computable, to arrive at optimal solutions for the dwell time problem. We extended the dwell time constraints to a more general constraints on the sequence of modes and considered the application of our results for the general problem to solve problems in a diverse area of precision agriculture, to demonstrate the wide applicability and importance of the results presented in this thesis.

We considered the problem of co-optimizing motion and communication energy in power aware mobile networks in realistic setting and a number of constraints. The co-optimization has the structure of hybrid optimal control in the objective function as well as the dynamics and we presented methods first for solving the problem in continuous domain and then as truly hybrid optimal control problem. The problem was solved in the setting of real-time situation by performing online channel prediction and subsequent optimization and the effectiveness of the schemes for computing the optimal controls was demonstrated by application examples. While the implementation is discrete, we solve the problem in continuous time rather than

discretizing the problem itself which allows us to use the Maximum Principle and hence our Hamiltonian based algorithm for rapid computation of optimal controls. A future work in this direction can be to extend our framework to solve other interesting problems in multi-agent networks required to perform co-ordinated tasks on the go

The problem of dwell time constraints was considered for the case of optimal mode scheduling for hybrid dynamical systems, with the continuous u being absent. A future work in this direction is to extend it to the case of *Full Optimization* where optimization is performed over schedules as well as continuous input while respecting the dwell time constraints. A nice application can be to hybrid co-optimization problem considered in chapter 8, since the switching of spectral efficiency is also subject to dwell times, albeit small, and would be amenable to solution in this extended framework.

We considered a multiple shooting technique that can be very effective for solving a class of optimal and hybrid optimal control problems. Our approach uses multiple shooting method in conjunction with gradient descent for solving a two point boundary problem resulting from the optimal control problem. The technique has the promise of solving complex optimal control problems over large time horizons. Solving hybrid optimal control problems with continuous as well as switching controls using suitable modification of our approach is a worthwhile opportunity for further investigations.

REFERENCES

[1] Ali, U., Cai, H., Mostofi, Y., and Wardi, Y., "Motion-communication co-optimization in mobile robotics: an optimal control perspective," *in preparation for submission to IEEE Transactions on Control of Networked Systems (TCNS)*.

[2] Ali, U., Cao, H., Mostofi, Y., and Wardi, Y., "Motion and communication co-optimization with path planning and online channel prediction," in *American Control Conference (ACC), 2016*, IEEE, 2016.

[3] Ali, U. and Egerstedt, M., "Hybrid optimal control under mode switching constraints with applications to pesticide scheduling," *submitted to ACM Transactions on Cyber Physical Systems (TCPS)*.

[4] Ali, U. and Egerstedt, M., "Optimal control of switched dynamical systems under dwell time constraints: Theory and computations," *in preparation for submission to IEEE Transactions on Automatic Control (TAC)*.

[5] Ali, U. and Egerstedt, M., "Optimal control of switched dynamical systems under dwell time constraints," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 4673–4678, Dec 2014.

[6] Ali, U., Yan, Y., Mostofi, Y., and Wardi, Y., "An optimal control approach for communication and motion co-optimization in realistic fading environments," in *American Control Conference (ACC), 2015*, pp. 2930–2935, IEEE, 2015.

[7] Ali, U. and Wardi, Y., "A multiple shooting technique for solving optimal control problems with application to power aware networks," in *Analysis and Design of Hybrid Systems (ADHS), 2015*, pp. 2930–2935, IFAC, 2015.

[8] Antsaklis, P. J., "A brief introduction to the theory and applications of hybrid systems," in *Proc IEEE, Special Issue on Hybrid Systems: Theory and Applications*, Citeseer, 2000.

[9] Armijo, L., "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[10] Axelsson, H., Wardi, Y., Egerstedt, M., and Verriest, E., "Gradient descent approach to optimal mode scheduling in hybrid dynamical systems," *Journal of Optimization Theory and Applications*, vol. 136, no. 2, pp. 167–186, 2008.

[11] Bellman, R. E., "Dynamic programming," *Cambridge Studies in Speech Science and Communication. Princeton University Press, Princeton*, 1957.

[12] Bengea, S. C. and DeCarlo, R. A., "Optimal control of switching systems," *Automatica*, vol. 41, no. 1, pp. 11 – 27, 2005.

[13] Bertsekas, D. P., *Nonlinear programming*. Athena Scientific, 1999.

[14] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[15] Branicky, M. S., "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 43, no. 4, pp. 475–482, 1998.

[16] Branicky, M., Borkar, V., and Mitter, S., "A unified framework for hybrid control: model and optimal control theory," *Automatic Control, IEEE Transactions on*, vol. 43, pp. 31–45, Jan 1998.

[17] Bryson, A. E. and Ho, Y.-C., *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[18] Caines, P. and Shahid Shaikh, M., "Optimality zone algorithms for hybrid systems computation and control: From exponential to linear complexity," in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pp. 1292–1297, June 2005.

[19] Caines, P. E. and Shaikh, M. S., "Optimality zone algorithms for hybrid systems: Efficient algorithms for optimal location and control computation," in *Hybrid Systems: Computation and Control*, pp. 123–137, Springer, 2006.

[20] Caldwell, T. and Murphey, T., "An adjoint method for second-order switching time optimization," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 2155–2162, Dec 2010.

[21] Caldwell, T. and Murphey, T., "Single integration optimization of linear time-varying switched systems," *Automatic Control, IEEE Transactions on*, vol. 57, pp. 1592–1597, June 2012.

[22] Caldwell, T. and Murphey, T., "Projection-based optimal mode scheduling," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 5307–5314, Dec 2013.

[23] Chatterjee, D. and Liberzon, D., "Stability analysis of deterministic and stochastic switched systems via a comparison principle and multiple lyapunov functions," *SIAM J. Control Optim.*, vol. 45, pp. 174–206, Jan. 2006.

[24] Ding, X., Wardi, Y., and Egerstedt, M., "On-line optimization of switched-mode dynamical systems," *Automatic Control, IEEE Transactions on*, vol. 54, pp. 2266–2271, Sept 2009.

[25] Egerstedt, M., Wardi, Y., and Axelsson, H., "Transition-time optimization for switched-mode dynamical systems," *Automatic Control, IEEE Transactions on*, vol. 51, pp. 110–115, Jan 2006.

[26] Egerstedt, M., Wardi, Y., and Delmotte, F., "Optimal control of switching times in switched dynamical systems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2138–2143, IEEE, 2003.

[27] Egerstedt, M., Azuma, S.-i., and Wardi, Y., "Optimal timing control of switched linear systems based on partial information," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 65, no. 9, pp. 1736–1750, 2006.

[28] El-Moukaddem, F., Torng, E., Xing, G., and Xing, G., "Mobile relay configuration in data-intensive wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 2, pp. 261–273, 2013.

[29] Filippov, A. F., "Differential equations with discontinuous right-hand side," *American Mathemtical Society Transl.*, vol. 42, pp. 199–231, 1964.

[30] Flint, M. L., *IPM in practice: principles and methods of integrated pest management*, vol. 3418. UCANR Publications, 2012.

[31] Georgakakos, A., Yao, H., Kistenmacher, M., Georgakakos, K., Graham, N., Cheng, F.-Y., Spencer, C., and Shamir, E., "Value of adaptive water resources management in northern california under climatic variability and change: Reservoir management," *Journal of Hydrology*, vol. 412-413, pp. 34 – 46, 2012.

[32] Ghaffarkhah, A. and Mostofi, Y., "Communication-aware motion planning in mobile networks," *IEEE Transactions on Automatic Control, special issue on Wireless Sensor and Actuator Networks*, vol. 56, no. 10, pp. 2478–2485, 2011.

[33] Giua, A., Seatzu, C., and Van Der Mee, C., "Optimal control of switched autonomous linear systems," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3, pp. 2472–2477, IEEE, 2001.

[34] Goldenberg, D. K., Lin, J., Morse, A. S., Rosen, B. E., and Yang, Y. R., "Towards mobility as a network control primitive," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 163–174, ACM, 2004.

[35] Goldsmith, A., *Wireless communications*. Cambridge university press, 2005.

[36] Gonzalez, H., Vasudevan, R., Kamgarpour, M., Sastry, S., Bajcsy, R., and Tomlin, C., "A numerical method for the optimal control of switched systems," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 7519–7526, Dec 2010.

[37] Gonzalez, H., Vasudevan, R., Kamgarpour, M., Sastry, S. S., Bajcsy, R., and Tomlin, C. J., "A descent algorithm for the optimal control of constrained nonlinear switched dynamical systems," in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pp. 51–60, ACM, 2010.

[38] Hale, M., Wardi, Y., Jaleel, H., M.Egerstedt, and Ali, U., "Hamiltonian-based algorithm for optimal control," *in preparation for submission to IFAC Nonlinear Analysis: Hybrid Systems (NAHS).*

[39] Hedlund, S. and Rantzer, A., "Optimal control of hybrid systems," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 4, pp. 3972–3977 vol.4, 1999.

[40] Hedlund, S. and Rantzer, A., "Convex dynamic programming for hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 47, no. 9, pp. 1536–1540, 2002.

[41] Hespanha, J. P. and Morse, A. S., "Stability of switched systems with average dwell-time," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3, pp. 2655–2660, IEEE, 1999.

[42] Howitt, R., Medellín-Azuara, J., MacEwan, D., Lund, J., and Sumner, D., "Economic analysis of the 2014 drought for california agriculture," tech. rep., UC Davis Center for Watershed Sciences and ERA Economics, 2014. Prepared for California Department of Food and Agriculture.

[43] Jaleel, H., Wardi, Y., and Egerstedt, M., "Minimizing mobility and communication energy in robotic networks: An optimal control approach," in *American Control Conference (ACC), 2014*, pp. 2662–2667, IEEE, 2014.

[44] Johnson, E. R. and Murphey, T. D., "Second-order switching time optimization for nonlinear time-varying dynamic systems," *Automatic Control, IEEE Transactions on*, vol. 56, no. 8, pp. 1953–1957, 2011.

[45] Jones, A., Ali, U., and Egerstedt, M., "Optimal pesticide scheduling in precision agriculture," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 1–8, April 2016.

[46] Liberzon, D., *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.

[47] Little, E., Brannen, P., Cline, B., Bost, S., Ahmad, A. S., Hale, F., Burrack, H., Liburd, O., Bessin, R., Czarnota, M., Jennings, K., Lockwood, D., and Smith, P., "2015 southeast regional organic blueberry pest management guide."

[48] Liu, X., Zhang, K., Li, S., Fei, S., and Wei, H., "Optimal timing control of switched stochastic systems," *IMA Journal of Mathematical Control and Information*, p. dnu013, 2014.

[49] Malmirchegini, M. and Mostofi, Y., "On the spatial predictability of communication channels," *IEEE Trans. on Wireless Comm.*, vol. 11, pp. 964–978, Mar. 2012.

[50] McCready, M., Dukes, M., and Miller, G., "Water conservation potential of smart irrigation controllers on st. augustinegrass," *Agricultural Water Management*, vol. 96, no. 11, pp. 1623 – 1632, 2009.

[51] Morse, A., "Supervisory control of families of linear set-point controllers part i. exact matching," *Automatic Control, IEEE Transactions on*, vol. 41, pp. 1413–1431, Oct 1996.

[52] Mostofi, Y., Malmirchegini, M., and Ghaffarkhah, A., "Estimation of communication signal strength in robotic networks," in *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pp. 1946–1951, May 2010.

[53] Ojiambo, P. S. and Scherm, H., "Survival analysis of time to abscission of blueberry leaves affected by septoria leaf spot," *Phytopathology*, vol. 95, pp. 108–113, January 2005.

[54] Ooi, C. C. and Schindelhauer, C., "Minimal energy path planning for wireless robots," *Mobile Networks and Applications*, vol. 14, pp. 309–321, January 2009.

[55] Piccoli, B., "Hybrid systems and optimal control," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 1, pp. 13–18 vol.1, 1998.

[56] Piccoli, B., "Necessary conditions for hybrid optimization," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 1, pp. 410–415, IEEE, 1999.

[57] Polak, E., *Optimization: algorithms and consistent approximations*. Springer-Verlag New York, Inc., 1997.

[58] Pontryagin, L. S., Mishchenko, E., Boltyanskii, V., and Gamkrelidze, R., *The mathematical theory of optimal processes*. Wiley, 1962.

[59] Rantzer, A. and Johansson, M., "Piecewise linear quadratic optimal control," *Automatic Control, IEEE Transactions on*, vol. 45, no. 4, pp. 629–637, 2000.

[60] Rao, A. V., "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[61] Riedinger, P., Kratz, F., Iung, C., and Zanne, C., "Linear quadratic optimization for hybrid systems," in *IEEE Conference on Decision and Control*, vol. 3, pp. 3059–3064, IEEE; 1998, 1999.

[62] Rosen, J. B., "The gradient projection method for nonlinear programming. part i. linear constraints," *Journal of the Society for Industrial & Applied Mathematics*, vol. 8, no. 1, pp. 181–217, 1960.

[63] Rothlauf, F., *Design of modern heuristics: principles and application*. Springer, 2011.

[64] Scherm, H., Savelle, A. T., and Pusey, P. L., "Interactions Between Chill-Hours and Degree-Days Affect Carpogenic Germination in Monilinia vaccinii-corymbosi," *Phytopathology*, vol. 91, pp. 77–83, January 2001.

[65] Shahid Shaikh, M. and Caines, P., "On the hybrid optimal control problem: Theory and algorithms," *Automatic Control, IEEE Transactions on*, vol. 52, pp. 1587–1603, Sept 2007.

[66] Shaikh, M. S. and Caines, P. E., "On trajectory optimization for hybrid systems: theory and algorithms for fixed schedules," in *IEEE Conference on Decision and Control*, vol. 2, pp. 1997–1998, IEEE; 1998, 2002.

[67] Shaikh, M. and Caines, P., "On the optimal control of hybrid systems: analysis and zonal algorithms for trajectory and schedule optimization," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2144–2149 Vol.3, Dec 2003.

[68] Slaughter, D., Giles, D., and Downey, D., "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 63 – 78, 2008. Emerging Technologies For Real-time and Integrated Agriculture Decisions.

[69] Sussmann, H. J., "Set-valued differentials and the hybrid maximum principle," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 1, pp. 558–563, IEEE, 2000.

[70] Sussmann, H., "A maximum principle for hybrid optimal control problems," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 1, pp. 425–430 vol.1, 1999.

[71] Tang, C. and McKinley, P. K., "Energy optimization under informed mobility," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, no. 9, pp. 947–962, 2006.

[72] Tokekar, P., Karnad, N., and Isler, V., "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, pp. 1–22, 2013.

[73] United States Environmental Protection Agency (EPA), Washington, D.C., *How to Comply With the Worker Protection Standard For Agricultural Pesticides: What Employers Need to Know*, September 2005.

[74] United States Environmental Protection Agency (EPA), Washington, D.C., *Federal Insecticide Fungicide and Rodenticide Act (FIFRA) 7 U.S.C. ch. 6*, 2012.

[75] Van Der Schaft, A. J., Schumacher, J. M., van der Schaft, A. J., and van der Schaft, A. J., *An introduction to hybrid dynamical systems*, vol. 251. Springer London, 2000.

[76] Wardi, Y. and Egerstedt, M., "Algorithm for optimal mode scheduling in switched systems," in *American Control Conference (ACC), 2012*, pp. 4546–4551, June 2012.

[77] Wardi, Y., Egerstedt, M., and Hale, M., "Switched-mode systems: gradient-descent algorithms with armijo step sizes," *Discrete Event Dynamic Systems*, pp. 1–29, 2014.

[78] Wardi, Y., Egerstedt, M., and Twu, P., "A controlled-precision algorithm for mode-switching optimization," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 713–718, Dec 2012.

[79] Xu, X. and Antsaklis, P. J., "Optimal control of switched autonomous systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 4, pp. 4401–4406, IEEE, 2002.

[80] Xu, X. and Antsaklis, P. J., "Optimal control of switched systems via nonlinear optimization based on direct differentiations of value functions," *International Journal of Control*, vol. 75, no. 16-17, pp. 1406–1426, 2002.

[81] Xu, X. and Antsaklis, P. J., "Results and perspectives on computational methods for optimal control of switched systems," in *Hybrid Systems: Computation and Control*, pp. 540–555, Springer, 2003.

[82] Xu, X. and Antsaklis, P., "Optimal control of switched systems based on parameterization of the switching instants," *Automatic Control, IEEE Transactions on*, vol. 49, pp. 2–16, Jan 2004.

[83] Yan, Y. and Mostofi, Y., "To go or not to go: On energy-aware and communication-aware robotic operation," *Control of Network Systems, IEEE Transactions on*, vol. 1, pp. 218–231, Sept 2014.

[84] Yan, Y. and Mostofi, Y., "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 4, pp. 1562–1572, 2013.

[85] Yu, S. and Lee, C. G., "Mobility and routing joint design for lifetime maximization in mobile sensor networks," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 2300–2305, IEEE, 2011.

[86] Yu, S., Lee, C., and Hu, J., "Energy optimal control in mobile sensor networks using hybrid systems theory," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pp. 1975–1980, Oct 2011.